NASA CONTRACTOR REPORT

NASA CR-86

NASA CR-86

ADAPTIVE-PREDICTIVE MODELING OF NONLINEAR PROCESSES

by P. M. DeRusso, R. J. Roy, R. W. Miller, and B. W. Nutting

Prepared under Grant No. NsG-244-62 *by*
RENSSELAER POLYTECHNIC INSTITUTE
Troy, New York
for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1964

ADAPTIVE-PREDICTIVE MODELING

OF NONLINEAR PROCESSES

By P. M. DeRusso, R. J. Roy, R. W. Miller,
and B. W. Nutting

Distribution of this report is provided in the interest of information exchange and should not be construed as endorsement by NASA of the material presented. Responsibility for the contents resides in the author or organization that prepared it.

Prepared under Grant No. NsG-244-62 by
RENSSELAER POLYTECHNIC INSTITUTE
Troy, N.Y.

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Office of Technical Services, Department of Commerce,
Washington, D.C. 20230 -- Price $3.00

TABLE OF CONTENTS

I. Introduction

1.1 Problem Description and Importance

The physical processes which exist are, for the most part, controllable. In order for man to exert control over these processes however, it is necessary for him to design a system to suitably perform this task. Such a design requires a characterization of the process to be controlled.

Often the characterization of a process can be determined from the physical principles involved in the process behavior. This has been true for most of the control systems designed in the past. As man seeks to control more complicated processes however, it is found that either many processes are extremely difficult to physically characterize, or that the characterization is too complicated to use in the determination of a suitable control system. For a complex process of high order with several non-linearities, there is no procedure which can accurately obtain all the process parameters. Fortunately however, there is no reason for obtaining all the process parameters and all the nonlinear functionalities.

The basic question of process identification is cause and effect. For a given cause, what is the effect. For example, in order to control human response, it is not required that the complex nerve and brain structure be known. The only practical method would be to observe a series of causes and effects, and from this data infer the proper cause to produce a desired effect. Not only is this procedure practical, but it is adaptive as well. If the cause-effect relationship changes, the observer is made aware of this fact. A parameter tracking for a structure as complex as a human being would be absurd. It is equally as absurd to track the parameters of a tenth order nonlinear process.

For these reasons, a relatively simple and straightforward method of relating process inputs and outputs is necessary. This method should be suitable for nonlinear, as well as linear, processes. Furthermore the process characterization should adaptively follow process changes due to such things as changes in environment. Since special test signals result in a disturbance of the process outputs, they should not be required to characterize the process.

Finally, it should be noted that for control purposes one is interested in determining an input to produce a desired output. It is important to realize this does not mean it is necessary to determine the process transfer function or describing function, or differential equations, etc. In fact, some of the above mentioned types of characterizations are not applicable to many processes of interest.

1.2 Past Work

Much has been written on the identification of process dynamics for the linear case.[1,2] As such however, it forms a very narrow view of a general problem. With the exception of the work covered in this report, the general problem has for the most part, been ignored.

1.3 Summary

The key to the research herewith reported is the recognition of two basic facts. The first of these is that one is interested in process identification for purposes of controlling the process. Thus what is desired is a knowledge of the inputs required to produce desired outputs. Exact process transfer functions parameter values, etc. are unnecessary.

The second fact is that determination of the inputs required to produce specific outputs is a pattern recognition problem. The input signals are analogous to patterns in time. If one has a table of possible process input patterns and the corresponding outputs which result, the process is characterized. Furthermore, it is then a relatively straightforward matter to determine the input patterns or signals necessary to control the process in a desired fashion.

Thus the seemingly widely separated problems of process and signal identification and prediction, molar psychological models of learning, pattern recognition, specimen classification, etc. are in reality common problems. Learning theory is a concept which unifies these separate problems and provides a solution.

The process which has a switched two-level input signal is considered in Chapter II of this report. This type of process is considered first because it provides a conceptually clear introduction to the approach, and because of the developing interest in predictive systems and the wide utilization of on-off control as practical solutions to least time optimization problems.

The theory for the switched two-level input signal process is presented along with an analysis of the memory capacity and time required to characterize such a process. A suggested computer structure for the process model is presented, along with a discussion of a constructed model and the experimental results obtained. The use of the model for control purposes is also discussed.

Chapter III is a similar presentation, but is not restricted to the switched two-level input signal process. This more general case, if handled in the same manner as the two-level case, would result in impractical computer requirements. Thus the learning theory approach is introduced to circumvent this problem and yield a practical method for process identification.

Conclusions are presented at the end of Chapters 2 and 3 for the respective cases. In general, the experimental tests and computer studies verify the theory and indicate that practical, useful identifiers can be constructed and utilized.

II Switched Two-Level Input Processes

2.1 Introduction

This Chapter presents a method by which the class of nonlinear processes with switched two level inputs and finite settling times can be identified and an adaptive model of the process constructed. The model uses the process input-output records, storing the input in a digital shift register tapped at n points, thus forming a function space of 2^n nonoverlapping cells. By averaging the output waveform during the time that a cell is occupied, a coefficient is obtained which characterizes the output for that input condition. It is indicated that 50-200 coefficients can be utilized to reasonably model the process. Thus the memory requirements are not excessive. This and the use of serial access result in a practical model.

The theory is presented in sufficiently general form to encompass the problem of prediction of the output of a process. The prediction involves an identification of both the process and its input signal. Thus the predictor must be used in an open loop fashion, such as for prediction of the effects of disturbances about a nominal trajectory.

The model is a small synchronous digital computer which is quite versatile, since its operation is independent of the type of process nonlinearity, and since it can adapt to systems with different settling times. It is also adaptive in the sense that it will follow slow changes in the process dynamics.

It appears that this type of model is useful for identification of slowly varying processes, if the parameter variations are constant over a time duration equal to approximately 100 process settling times. Prediction increases the stationarity requirements on the process parameters by a factor of five to ten.

The importance of the switched two-level process has its foundation in the utilization of predictive control as a practical solution to least time optimization problems.[3,4] Chestnut et al. indicate significant applications for predictive control in space navigation and rendezvous missions, aircraft landing problems, and the control of chemical plants.[4] The process identification concepts presented in this chapter are ideally suited for predictive control applications.

2.2 Delay Line Synthesizer

One of the difficulties in obtaining a model of a process is the fact that the process is "on-line". Therefore the model must be obtained by examination of the input-output operating record. For a linear, time invariant system the input-output relationship is given by the convolution integral

$$y(t) = \int_0^\infty h(\tau) \quad x(t-\tau) \quad d\tau \qquad (2\text{-}1)$$

where $y(t)$ = output of the system

$x(t)$ = input to the system

$h(t)$ = system impulse response

Given the input-output relationship, the problem is one of "deconvolving" the integral expression to obtain $h(t)$, the system impulse response. Delay line synthesis is an approximate technique for "deconvolving" the integral expression for systems which have a finite impulse response.[5]

The Delay Line model consists primarily of a tapped delay line. The outputs from the taps are weighted and summed to form the output of the model. The output of the Delay Line Synthesizer is given by

$$y(t) = \sum_{n=o}^{k} a_n x (t-nT)$$

where

a_n = weight of n^{th} tap

T = delay time between taps

If the weights a_n are adjusted so that

$$a_n = T h(nT)$$

then the output is given by

$$y(t) = T \sum_{n=o}^{k} h(nT) \ x (t-nT)$$

This is a discrete approximation to the convolution integral of Eq. 2-1.

The coefficients a_1, a_2, --- a_K are adjusted by cycling the process input operating record into the model and adjusting the coefficients to obtain the best fit of the model output to the process output operating record. This is an iterative technique and requires many adjustments of each coefficient before the model is final. It is necessary to have the input-output data recorded so that it can be played back into the adjusting apparatus.

The contamination of the normal operating record by random noise is overcome by the use of correlation functions. The relationship

$$\phi_{xy} (\tau) = \int_{0}^{\infty} \phi_{xx} (\tau - \lambda) \; h \; (\lambda) \; d \qquad (2\text{-}2)$$

where $\phi_{xy} (\tau)$ = input-output cross correlation function

$\phi_{xx} (\tau)$ = input autocorrelation function

provides the means for reducing the coefficient uncertainty due to random noise inputs. Note that Eq. 2-2 is quite similar to Eq. 2-1. If the input record is replaced by the input autocorrelation record and the output record by the input-output cross correlation record, then the inputs to the adjusting setup (the correlation records) will be essentially noise free.

For a proper Delay Line model it can be shown that the minimum length of the delay line should be as long as P (the reciprocal of the bandwidth of the process) for overdamped systems and between 1.5P and 2P for underdamped systems. The length of the delay element T should be less than 1/10 the reciprocal of the system bandwidth, i.e.

$$T/P < 0.1$$

A more efficient use of the tap spacing is where the delay line is tapered.[6] The principal information about an impulse response is contained in the first half of the impulse response. A tapered delay line with the taps closely spaced at the beginning of the line satisfies this requirement.

The ideas concerning the Delay Line Synthesizer will be utilized in evolving a model for a nonlinear plant. The principal concept of the Delay Line Synthesizer is that by an appropriate linear transformation on a finite set of past input points, a model of a linear system can be obtained which approximates the linear system in the mean.

2.3 Wiener-Bose Theory of Nonlinear Systems

The output of a linear system can be expressed in terms of the input to the system and the system weighting function (the impulse response) by means of the convolution integral. For nonlinear systems there is no direct relationship between input and output, as the principle of superposition does not hold.

The most promising approach upon which to base a modeling theory for nonlinear systems is that taken by Wiener and Bose.[7] Wiener showed that any system can be regarded as a computer which performs a transformation on the past of its input to yield the present output. For the case where the transformation is linear, one can make use of the convolution integral to obtain the present output from the past of the input. The linear system is then characterized by its response to a unit impulse. This is the basis for the Delay Line Synthesizer. The Delay Line Synthesizer "deconvolves" the convolution integral to obtain the system impulse response.

The principal contribution of the Wiener theory is that it shows that any nonlinear system can be characterized by a linear network with multiple outputs cascaded with a nonlinear network with no memory of the past (zero memory device). The multiple output linear network serves to characterize the past of the input. The nonlinear network

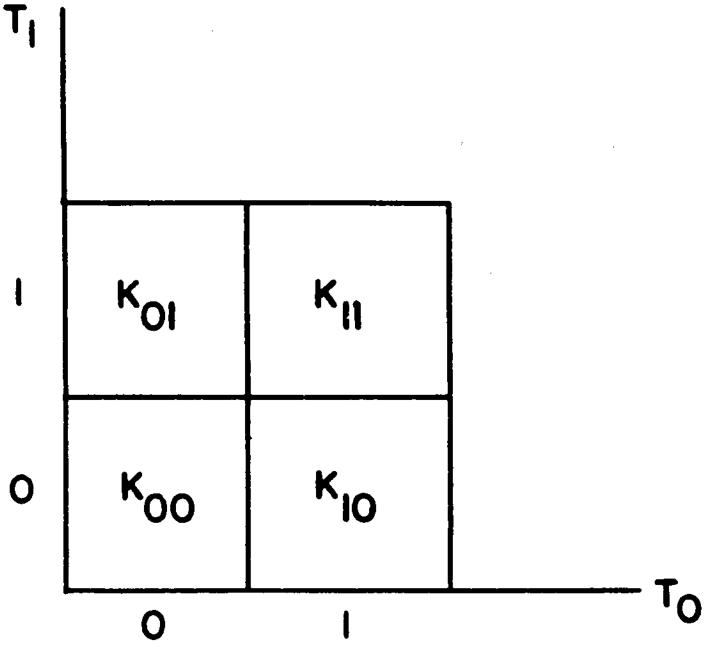


FIGURE 2-1

operates on this information to yield the present output. This chapter uses the Wiener-Bose concept to obtain a practical solution to the identification and modeling of nonlinear systems with two-level inputs.

2.4 Theory of the Model

It is assumed that the input signal switches between two levels. In the remaining discussion, the lower level is designated as the 0 level, and the higher level as the 1 level. The input past of interest can be stored in a shift register having n tap points. Each tap is designated by the symbol T_K, where K represents the number of delay units from the input. The notation $T_K(0)$ indicates that the level at the K^{th} tap is at the lower level. The notation $T_K(1)$ indicates that the level at the K^{th} tap is at the higher level.

There can exist 2^n different binary sequences on the shift register. Since only one level can exist at a tap point at any one time, only one of these 2^n sequences can exist at any one time. By assigning a coefficient to each of the 2^n possible sequences, the model output Z(t) can be expressed as an expansion, where only one term in the expansion can exist at a given time.

As a simple example of this principle, consider the case where there are only two taps. There are four possible binary sequences that can exist on the shift register. These four conditions can be shown on a T_0-T_1 plane, as shown in Figure 2-1. These four conditions (or cells) are non-overlapping. Therefore a unique coefficient can be assigned to each cell. The model output Z(t) is determined by the cell location of the input past, i.e.

$$Z(t) = K_{00}\ T_0(0)\ T_1(0) + K_{01}\ T_0(0)\ T_1(1)$$

$$+ K_{10}\ T_0(1)\ T_1(0) + K_{11}\ T_0(1)\ T_1(1)$$

$$Z(t) = \sum_{i=0}^{1} \sum_{j=0}^{1} K_{ij}\ T_o(i)\ T_1(j)$$

Only one term in this expansion is non-zero at any one time. For example, the output is equal to K_{11} when the input is at the 1 level on tap T_o and tap T_1.

From this simple example, the general structure of the method is seen. The input past is divided into 2^n cells, where each cell is non-overlapping and represents a unique input past. Each cell is assigned a coefficient. When the input past appears in one of the cells, the model output Z(t) is the value of the coefficient assigned to that cell. In general then, the output can be expressed as

$$Z(t) = \sum_{i=0}^{1} \sum_{j=0}^{1} \cdots\cdots \sum_{z=0}^{1} K_{ij\ \ldots\ z}\ T_0(i)\ T_1(j)\ \cdot\cdot T_{n-1}(z)$$

where there are n summations.

The above expression can be simplified (in form) since each term in the expansion is mutually exclusive with respect to the other terms. Let

$S(\alpha)$ = sequence α, one of the possible sequences described by

$$T_0(i)\ T_1(j)\ \ldots\ T_{n-1}(z)$$

where i, j ... z take on the values zero or one. Further, let $K_{ij\ldots z} = K_{\alpha}$. The output can then be expressed as the summation of 2^n mutually exclusive terms.

$$Z(t) = \sum_{\alpha} K_{\alpha} S(\alpha) \tag{2-3}$$

The term $S(\alpha)$ is <u>one</u> if the input past is in the α cell and <u>zero</u> if it is not.

The mutually exclusive nature of the sequences is expressed by Eq. 2-4.

$$\sum_{\alpha} S(\alpha)\, S(\beta) = S(\beta) \tag{2-4}$$

where $S(\alpha)$ is a switching function having the values zero or one. K_{α} is a constant chosen to minimize the mean square difference between y(t), the process output, and Z(t), the model output, assuming that the sequence α exists on the shift register at time t.

2.5 Determination of the Coefficients

Since the model described above could also be used for prediction of the process output, the derivation of the values of the K_{α}'s is presented in a sufficiently general form to encompass prediction. With this in mind, a superscript is added to the K_{α}'s to denote the instant of time, real (t) or future (t+T), for which they apply. Thus the constant K_{α}^{t+T} represents the best estimate of the future process output y(t+T), given a particular input past. The determination of K_{α}^{t+T} for the general case of identification or prediction is as follows:

The mean square error E is given by Eq. 2-5, where the bar indicates a time average

$$\overline{e(t)^2} = E = \overline{\left[y(t+T) - \sum_{\alpha} K_{\alpha}^{t+T} \; S(\alpha)\right]^2} \qquad (2\text{-}5)$$

Differentiating with respect to a dummy term K_β

$$\frac{\partial E}{\partial K_\beta} = \overline{-2 \left[y(t+T) - \sum_{\alpha} K_{\alpha}^{t+T} \; S(\alpha)\right] S(\beta)} \qquad (2\text{-}6)$$

Setting Eq. 2-6 equal to zero

$$\overline{y(t+T)\,S(\beta)} = \overline{S(\beta) \sum_{\alpha} K_{\alpha}^{t+T} \; S(\alpha)} \qquad (2\text{-}7)$$

Utilizing Eq. 2-4, the desired equation for K_β becomes

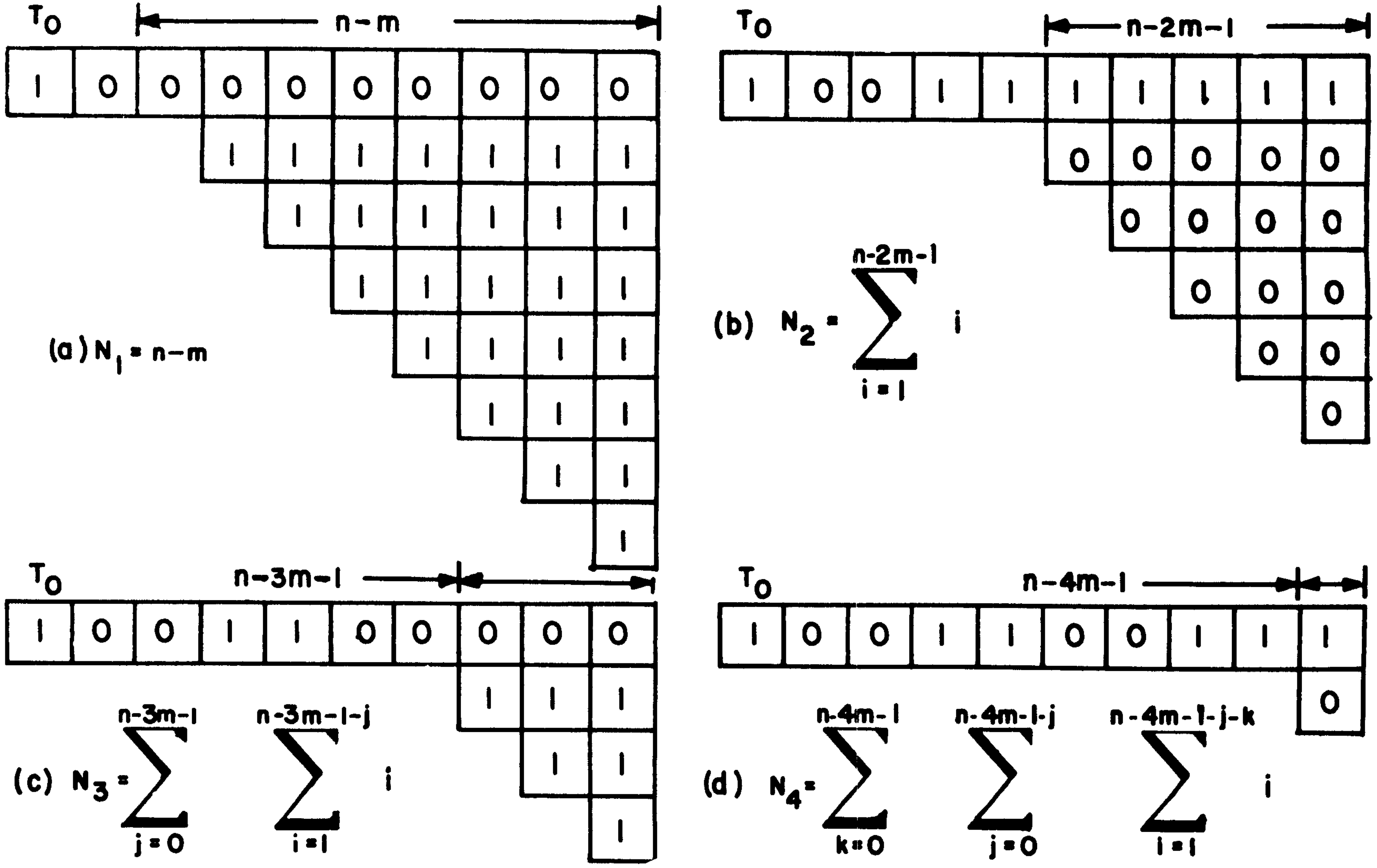
$$K_{\beta}^{t+T} = \frac{\overline{y(t+T)\,S(\beta)}}{\overline{S(\beta)}} \qquad (2\text{-}8)$$

The above expression for K_{β}^{t+T} states that the best estimate for $y(t+T)$ is the average value of $y(t+T)$, given the existence of $S(\beta)$ at time t.

The numerator of Eq. 2-8 represents a correlation function with time shift T. Since $S(\beta)$ has values of zero or one, the denominator $S(\beta)$ can be viewed as the probability that the input sequence is $S(\beta)$. The coefficient K_{β}^{t+T} can also be expressed as a conditional mean.

$$K_{\beta}^{t+T} = \left[Z(t) \;\middle|\; S_t(\beta)\right] = \left[\overline{y(t)} \;\middle|\; S_{t-T}(\beta)\right] \qquad (2\text{-}9)$$

where the latter form holds in the stationary case. $S_t(\beta)$ is the present



(a) $N_1 = n-m$

(b) $N_2 = \sum_{i=1}^{n-2m-1} i$

(c) $N_3 = \sum_{j=0}^{n-3m-1} \sum_{i=1}^{n-3m-1-j} i$

(d) $N_4 = \sum_{k=0}^{n-4m-1} \sum_{j=0}^{n-4m-1-j} \sum_{i=1}^{n-4m-1-j-k} i$

FIGURE 2-2

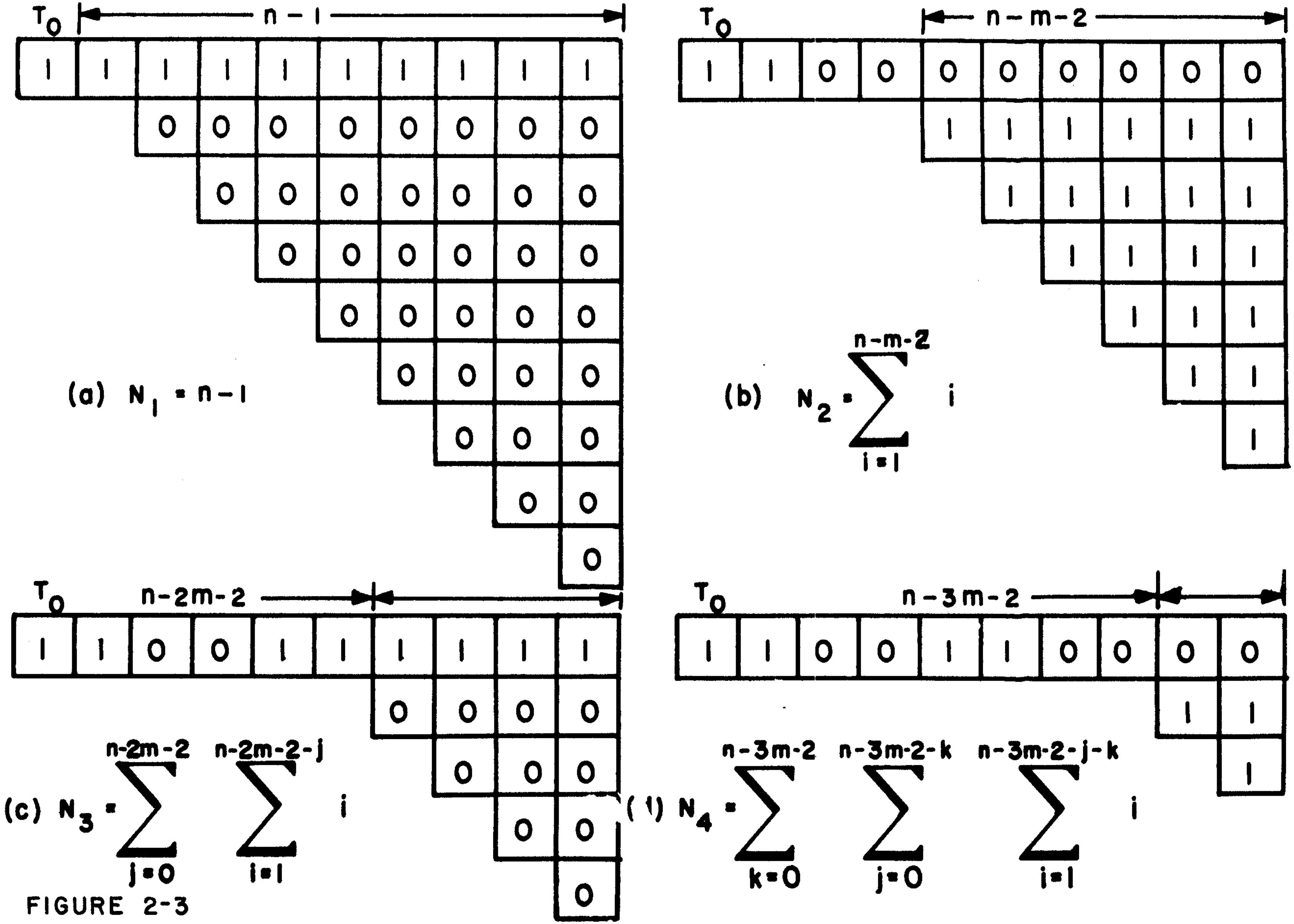
sequence, and $S_{t-T}(\beta)$ is the sequence that existed T units in the past.

Essentially the prediction process is a memory process. The model stores the average value of the present output for the input sequence that existed T units in the past. When that particular input sequence occurs again, the model "remembers" what the average value of the subsequent output was when that sequence appeared previously, and predicts this value for the future model output.

2.6 Expected Number of Coefficients

A sequence of ten bits is sufficient to represent the memory characteristics of most two-level processes. For a ten tap input memory the number of coefficients required is 1024. The assumption underlying this number is that the input can switch as fast as $(\frac{1}{n})^{th}$ of the effective settling time of the system, where n is the number of taps. If it is known _a priori_ that the input does not switch this rapidly (for control system applications it generally will not) the number of coefficients is drastically reduced. For most control system applications the output must experience a sizable change before the input switches, so it is reasonable to expect some minimum time between input switching.

Therefore, the number of sequences required for an n tap system is determined assuming the input is constrained to a minimum of m consecutive ones or zeroes on these taps. As an illustration, consider the case for $n = 10$ and $m = 2$. Either the first two bits are the same, or they differ. First assume that they are different and have the values shown in Figure 2-2. The total number of different



(a) $N_1 = n-1$

(b) $N_2 = \sum_{i=1}^{n-m-2} i$

(c) $N_3 = \sum_{j=0}^{n-2m-2} \sum_{i=1}^{n-2m-2-j} i$

(d) $N_4 = \sum_{k=0}^{n-3m-2} \sum_{j=0}^{n-3m-2-k} \sum_{i=1}^{n-3m-2-j-k} i$

FIGURE 2-3

sequences where either one or two input switches occurred is shown in Figure 2-2a. In Figure 2-2b the number of different sequences where three input switches occurred is determined by adding an additional switch to each of the cases of Figure 2-2a where two input switches occurred. The number of different sequences where four input switches occurred (Figure 2-2c) is determined by adding an additional switch to each of the cases of Figure 2-2b where three input switches occurred. Similarly Figure 2-2d shows the determination of the number of states where five input switches occurred.

In Figure 2-3 the same procedure is repeated, but the assumption is that the first two tap values are alike. Since 01 or 00 could have been chosen for the first two tap values, the total number of sequences is then twice the total number of different sequences given by Figures 2-2 and 2-3. The final expression is given by Eq. 2-10, with a tabulation of the number of coefficients for n = 10 and m varying from one to ten shown in Table 2-1.

Table 2-1

Number of Coefficients Required

m	1	2	3	4	5	6	7	8	9	10
N	1024	178	82	52	40	32	26	22	20	20

$$
\begin{aligned}
N = 4n-2m-2 &+ 2 \sum_{i=1}^{n-m-2} i \\
&+ 2 \sum_{j=0}^{n-2m-2} \sum_{i=1}^{n-2m-j-2} i \\
&+ 2 \sum_{k=0}^{n-3m-2} \sum_{j=0}^{n-3m-2-k} \sum_{i=1}^{n-3m-2-j-k} i \\
&+ \text{-----------------------------------} \\
&+ 2 \sum_{i=1}^{n-2m-1} i \\
&+ 2 \sum_{j=0}^{n-3m-1} \sum_{i=1}^{n-3m-1-j} i \\
&+ 2 \sum_{k=0}^{n-4m-1} \sum_{j=0}^{n-4m-1-k} \sum_{i=1}^{n-4m-1-j-k} i \\
&+ \text{-----------------------------------} \qquad (2\text{-}10)
\end{aligned}
$$

where N = number of coefficients

n = number of taps

m = minimum number of consecutive ones or zeroes. $(m \leq n - 1)$

For $m \leq n$ the number of coefficients is $2n$.

From the data in Table 2-1, it would seem that a figure of 50 - 200 coefficients is reasonable to expect. For $m = 2$ there can be five input switches within the length of the effective process settling time. This is fairly rapid switching, and would not generally be expected in most control system applications.

2.7 Identification Time

To evaluate the time it would take for the model to accumulate each of the coefficients K_{α}, the following problem should be considered: What is the probability that every possible input sequence has occurred at least once in an operating record of a certain length?

This is an extremely difficult problem, because each sequence occurs with a different probability, which depends on the sequences that occurred in the past. A much simpler problem, which is already solved, is: If all sequences are equally likely and statistically independent, what is the probability that a record of N sequences contains each possible sequence at least once?[8] This idealization provides a lower bound on the identification time, and will indicate how the identification time varies with the number of possible sequences.

The idealized problem is that of putting N balls (sequences) into q different cells (sequences). An approximation to the exact solution, valid for large values of q and N, is

$$P(q,N) \sim \exp \left\{ -q \exp(-N/q) \right\} \tag{2-11}$$

where P(q,N) is the probability that no cell is empty (i.e. all possible sequences have occurred.)[8]

In the model, the input sequence may be expected to change at least once in every time duration corresponding to a tap interval. If there are 10 taps on the shift register memory and the length of the shift register memory is equal to the effective settling time of the system T_s, then the number of samples which occur during k T_s is

$$N = 9(k-1) + 1 \qquad (2\text{-}12)$$

since one occurs during the first T_s, and nine each T_s thereafter. Table 2-2 lists the number of settling times (or lengths of the shift register memory) required in the input record to make $P(q,N) = 0.9$ for various values of q.

Also listed in Table 2-2 is k, the average value of the identification time. This is the average time for all q sequences to occur. It is computed from $\bar{N}$, the average number of sequences required to obtain all q sequences. $\bar{N}$ is given as[8]

$$\bar{N} \sim q \log q, \text{ for large } q \qquad (2\text{-}13)$$

The probability that all q sequences will occur in $\bar{N}$ sequences is about 0.37 for large q.

One occurrence of a particular sequence "α" may not be sufficient to determine the coefficient K_α exactly, because the relative times of the input zero crossings may vary. However, if the shift register memory is longer than the effective settling time of the system, and if the input can switch only at time intervals equal to the tap interval, then one occurrence of the sequence is sufficient. This is true since the time average computed by the model will be the same each time that sequences occurs. The assumption regarding settling time is reasonable, since the model must be designed in this way to ensure accuracy. The assumption regarding the interval between zero crossings is not so artificial as might first appear. If the model is to be used in conjunction with some device which controls the input based on some optimum desired output, then it is reasonable to assume discrete switching intervals limited to multiples of the sampling interval. There is no basis for any finer control over the switching interval.

Table 2-2

Identification Time and Prediction

Number of Input states (q)	Time* for Identification $P(n,N) = 0.9$ (k)	Expected Identification Time* ($\bar{k}$)	Time* to determine Prediction Coefficients $\langle E-E_{min} \rangle \leq 1\%$ of E_{min}
50	35	22	556
100	77	52	1,110
200	168	118	2,220
500	470	346	5,550
1,000	1,017	767	11,100

*Times are normalized to system settling time T_s

Even if the above restrictions are removed however, the mean square error (MSE) would still be small. The additional MSE induced by the inaccuracy of the coefficients is shown in Appendix I to be less than the minimum MSE (E_{min}) when averaged over all values of the inaccurate K_α's. For identification this is reassuring, because E_{min} will be small.

If the model is to be used for prediction, E_{min} does not approach zero for an infinite memory of past inputs. It does, however, approach the MSE for the optimum prediction based on the infinite past. Therefore, the model is an approximation to the best possible predicting filter. If the prediction is based on only one sample, an additional error equal to E_{min} may be intolerable.

If each of the coefficients is based on M samples of its corresponding sequence, Appendix I shows that the <u>additional</u> MSE induced because there are only M samples is $1/M^{th}$ of the minimum MSE. Thus, if the coefficient for prediction is based on 64 samples, the MSE is increased by less than 1.6%.

If all sequences are equally likely and contribute equally to the error, then the average MSE induced is $(q/N)\ E_{min}$. Thus, for 50 input sequences and 1% additional MSE due to a finite number of samples, about 5000 samples are required. This corresponds to about 556 times the system settling time according to Eq. 2-12. Several other values are listed in Table 2-2.

The results for identification time listed in Table 2-2 compare favorably with the results of Cooper and Lindenlaub's work on identification time of linear systems.[2] They studied several different methods of system identification and concluded that the identification time using these methods is of the order of 10 to 100 times the significant duration of the impulse response. One expects identification times for nonlinear systems to be somewhat longer than this, because superposition techniques cannot be used.

For an adaptive control system the identification should be based on as short a record as possible. If the process characteristics vary slowly enough, then the identification is a time sequence of time-invarient models. However, there is a conflict of requirements, since errors may be too large if the record is too short. Thus, it appears that this type of model is useful for identification of slowly varying processes with two level inputs, if the parameter variations are fairly constant over roughly 100 times the process settling time. This figure is based upon the values in Table 2 for 50 to 100 input states. For prediction, this type of model is useful for processes whose parameters are fairly constant over about 500 to 1,000 times the settling time.

2.8 Errors for the Table Lookup Method

In order to form an idea of the magnitudes of errors involved, let the system be linear and stationary with weighting function $h(\tau)$, and let the input process by x and the output y. The model samples the input at n intervals and obtains a representation x^*. of the input. Since x is binary x^* is represented as a binary number. There are 2^n different inputs. For each, the model output z is the average value of the output given the input representation x^*.

$$z = \int_{-\infty}^{\infty} y \; p(y/x^*) \; dy = \bar{y}^{x^*}$$

The mean square error (MSE) when x* is present is

$$E_{x*} = \overline{(y-z)^2}^{x*} = \overline{y^2}^{x*} - \left(\overline{y^{x*}}\right)^2$$

and the overall MSE is

$$E = \sum_{x*=0}^{2^n-1} E_{x*} \, P(x*)$$

Now, since the system is linear

$$y = \int_0^\infty h(\tau) \, x(t-\tau) \, d\tau$$

and

$$\left(\bar{y}^{x*}\right)^2 = \int_0^\infty h(\tau) \int_0^\infty h(\sigma) \, \overline{x(t-\tau)}^{x*} \, \overline{x(t-\sigma)}^{x*} \, d\sigma \, d\tau$$

Thus, if the weighting function and input statistics are known, then the mean square error can be evaluated. In Appendix II it is shown that, if the input is ± 1, and has zero crossings governed by a Poisson distribution, then the MSE is given by:

$$E = E_\ell + \sum_{k=1}^{N} E_k$$

where

$$E_\ell = \int_0^\infty \int_0^\infty h(I_n + \tau) \, h(I_n + \sigma) \, \phi_1(\tau, \sigma) \, d\tau \, d\sigma$$

$$E_k = \int_0^{T_k} \int_0^{T_k} h(I_{k-1} + \tau) \, h(I_{k-1} + \sigma) \, \phi_2(\tau, \sigma) \, d\tau \, d\sigma$$

and

$$\phi_1 (\tau, \sigma) = 2e^{-2a \max \{\tau, \sigma\}} \quad \sinh \left[2a \min \{\tau, \sigma\}\right]$$

$$\phi_2 (\tau, \sigma) = e^{-2a |\tau - \sigma|} - \frac{\cosh \left[2a \left(T_k - \tau - \sigma\right)\right] - e^{-2aT_k} \cosh \left[2a(\tau - \sigma)\right]}{\sinh 2 a T_k}$$

and

T_k = time between kth and k + 1^{th} taps

$$I_k = \sum_{i=1}^{k} T_k, \; I_o = 0$$

In appendix III this expression is evaluated for the case where the system has $h(t) = 3^{-t}$, a low-pass filter. The expression for the mean square errors has the form:

$$E = \sum_{j=1}^{n} e^{-2 \sum_{i=1}^{j-1} T_i} \; g(\tau_j) \; + Ke^{-2 \sum_{i=1}^{n} T_i}$$

where g is monotonic increasing. One may minimize E with respect to T_k by setting $\frac{\partial E}{\partial T_k} = 0$:

$$\frac{\partial}{\partial T_k} \; \dot{g}(T_k) = 2 \left[\sum_{i=k+1}^{n} e^{-2 \sum_{i=k}^{j-1} g(T_j)} - K \, e^{-2 \sum_{i=k}^{n} T_i} \right]$$

The important point is that the minimum value of T_k depends only upon $T_{k+1}, \ldots, T_n$ and not upon $T_1 \ldots, T_{k-1}$. Thus it is possible to minimize first with respect to T_n, then T_{n-1}, etc. The result gives the optimum tap intervals, which are listed in Table 2-3. The results listed in Table 2-3 indicate that:

1) The relative error decreases as the average switching interval increases, since there is less uncertainty as to the value of the input between taps.

2) The tap intervals further in the past should be made longer than those nearer the present, since these values of input have less effect than the more recent ones.

3) The total line length to be used depends upon the input statistics, since the sampling gives only the input value at discrete times.

4) Adding an additional sampling point (thus doubling the model complexity) reduces the relative error by a relatively small amount. One presumably reaches a point where the improvement in accuracy is not worth the added complexity involved in increasing the number of samples.

5) The accuracy attainable seems to be sufficient for many applications. This accuracy cannot be maintained for systems whose inputs are continuous without increasing the storage requirements beyond all reason. Therefore, it is necessary to develop methods of reducing the storage requirement and to accept some additional errors in return. Such methods are presented in Chapter 3.

2.9 Use of Model in Minimum Time, Terminal Control Applications

This and the following section consider the controlling a process which has been identified by the model. This section is devoted to the

Table 2-3

Minimum Errors and Optimum Tap Intervals

Two Level Input, Table Lookup Model

$\alpha = 2 \text{ x } \frac{\text{System T.C.}}{\text{Ave Switching Interval}}$		0.25	0.5	2.0	5.0
Length of Tap Interval (in system T.C.'s) from most recent (1) to furthest in the past (9)	1.	0.1659	0.1614	0.1409	0.1151
	2.	0.1865	0.1809	0.1556	0.1251
	3.	0.2130	0.2057	0.1740	0.1373
	4.	0.2483	0.2385	0.1973	0.1522
	5.	0.2976	0.2838	0.2280	0.1711
	6.	0.3041	0.3000	0.2705	0.1960
	7.	0.3791	0.3580	0.3000	0.2309
	8.	0.5012	0.4658	0.3387	0.2849
	9.	0.7304	0.6610	0.4398	0.3000
TOTAL LINE LENGTH (System T.C.'s)		3.0261	2.8551	2.2448	1.7126
RMS ERROR (PERCENT)		5.4	6.6	16.4	29.5
RMS ERROR if 1 Tap is Added		4.0	5.9	15.0	27.5

problem of hitting a fixed terminal point in least time. Optimum trajectory following is considered in the next section.

In a minimum time terminal control application, the present process input sequence $S(\alpha)$, and y_α, the corresponding process output are known. The objective is to determine an input sequence $S(\beta)$ which will take the process output closest to a known desired value y_d in least time.

Assuming that the process has been identified, it is possible to search the model for the coefficient K_β^t (and hence the model output) which is closest to y_d. Since K_β^t represents the average behavior of the process during a time interval T equal to the shift register tap spacing, the corresponding process output may or may not be y_d. It is in the mean square sense however, the closest output which can be obtained within the knowledge of the model. Once K_β^t has been determined from the model memory,. the corresponding sequence $S(\beta)$ which causes K_β^t can be read from memory. There may be more than one possible sequence. Thus the problem is one of changing the input sequence from $S(\alpha)$ to $S(\beta)$ in least time, where $S(\beta_i)$ denotes all the input sequences which lead to the output K_β^t.

As an illustration, consider the problem of switching from an initial sequence $S(\alpha) = 10011$ to any of the sequences $S(\beta_i)$, where the $S(\beta_i)$ are

$$S(\beta_1) = 00110$$

$$S(\beta_2) = 01010$$

$$S(\beta_3) = 01111$$

each of which produces the desired K_β^t. At time t + T, the input sequence will be ___1001, where the blank could be a zero or a one. This does not match the last four bits of any of the $S(\beta_i)$, so now consider time t + 2T. At this time the input sequence will be — — 100. Since this still does not match any of the $S(\beta_i)$, proceed to t + 3T. Now the input sequence is — — — 10. This matches both $S(\beta_1)$ and $S(\beta_2)$, so that either 001 or 010 will produce the best output from the process.

The proper input sequence is thus easily obtained. The sequence is the first m bits of the $S(\beta_i)$ whose last (n-m) bits match with the first (n-m) bits of $S(\alpha)$. If a match is obtained, but the first m bits of the $S(\beta_i)$ do not form a proper input sequence, then the process continues until a proper sequence is found. It is obvious that limiting the minimum time between input zero crossings increases the average transition time. With a limit on the input sequence there is less control on the plant input.

2.10 Optimum Trajectory Following

The use of the model in a control system to cause the process output to follow a desired function of time is a more difficult problem than the least time, terminal control application. There are several approaches to the problem. Three of these are discussed here.

The system model is essentially a table which gives the output for each input sequence. The input is denoted by a sequence of zeroes and ones n long, representing the input at n equal intervals T. The input may switch only at discrete intervals. The problem to be considered is as follows:

If the system is now in a particular input state, then which of all possible input sequences will minimize the performance index J, given by:

$$J = \sum_{i=1}^{N} h' (y_i; i)$$

where y_i is the output at time iT from present and h' is some functional?

Now y_i is a function of the input sequence at time iT, and so J may be rewritten as

$$J = \sum_{i=1}^{N} h\ (S_i, i)$$

The i^{th} input sample x_i is either zero or one. Now for the two-level model with given initial state S_o, there are 2^i possibilities for S_i if $i \leq n$, and 2^n possibilities otherwise. For any optimization procedure the functions h (S_i, i) must be computed for all allowable S_i, i = 1, ..., N. If h is a function of the state only, the problem is reduced.

A straightforward method of determining the S_i is to compute J for every possible input sequence, then compare to find the minimum. There are 2^n possible input sequences. For each of these, an N-term sum must be calculated. This gives $N2^n$ operations. In addition, the 2^N sums must be compared. This is done by comparing the first two, discarding the larger, and then comparing the smaller with the third sum, discarding the larger, etc. This requires 2^{n-1} subtractions. The result is $(N+1)\ 2^n-1$ arithmetic operations.

It is conceivable that the number of operations might be reduced by performing comparisons before the entire sum has been computed. That is, suppose that at some point the minimum J, of those so far computed, is known. Then, when calculating the next J, for the next input sequence, it might be advantageous to pause after the first couple terms of the sum, to compare the partial sum with the previous minimum J. If it is greater, then there is no point in continuing the sum, and the next input sequence is then considered. If not, several more terms are added and the result compared with the previous minimum, etc. This has great advantages, since in eliminating a particular input sequence after the kth step, all of the 2^{N-k} input sequences which are extensions of the first k steps of that sequence are also eliminated.

Another method which reduces the number of operations, does so at the expense of memory space. Rather than adding each of the N terms in the sum for J, partial sums may be computed and stored. Thus, one starts by computing the first two partial sums corresponding to 0 and 1 for the first input. Then one computes the partial sums corresponding to 00, 01, 10, 11 for the first two inputs, etc. At each stage there are 2^{i+1} additions, yielding

$$\sum_{i=1}^{N} 2i = 2^{N+1} - 2$$

additions. Combining this with 2^N-1 comparisons, this gives $3\ (2^N-1)$ arithmetic operations.

This is a significant improvement over the original scheme. Of course, the partial sums need not be computed in the order indicated above. For instance one might first compute all partial sums (and the final sum) for the input sequence (00000) (if N = 5). Then it is an easy matter to compute (10000) and compare. Next one might compute the (-1000) partial sum and compare it with the previous minimum. If it is greater than the previous minimum, one need not bother computing (01000) or (11000). Continuing in like manner, further reductions in computation time are possible.

Using a dynamic programming approach, there are two cases. Either $N > n$ or $N \leq n$, where n is the number of bits in the input sequence. Considering the latter case, one works backward in the normal manner, finding the minimum error function for each possible input sequence. One calculates this first t = (N-1)T and works backward to t = 0. At t = 0 there is only one possible input state (the given one), and at t = iT there are 2^i.possible input states. For each state, the minimum error function is found by making two additions and one comparison. Thus the number of operations is

$$\sum_{i=0}^{N-1} 3 \cdot 2^i = 3\ (2^N - 1)$$

Note that this is exactly the number of operations required in solving the problem by calculating partial sums and making no comparisons until the end!

For the case where $N > n$, the result is quite different, for the number of states to be considered at any stage is never larger than 2^{n-1}, which is half the total number of states. This reduction comes about because states such as (0110) and (0111), differing only in the first bit, must have the same minimum error function, corresponding in this case to (0011) or (1011). This duplicity never occurs for $n \geq N$.

Now for the first n time intervals, there are $3(2^n-1)$ operations. But for each of the N-n additional time intervals, there are only $3 - 2^{n-1}$ operations, yielding a total of $3 \cdot 2^n + (N-n)\ 2^{n-1} - 1$ operations. Finally, dynamic programming can be applied to solve the problem for an arbitrary initial input sequence, a factor not to be overlooked. To do this would require $3N \cdot 2^{n-1}$ operations. Such a procedure might aid in determining optimal steady state behavior in following a D.C. level or a periodic function.

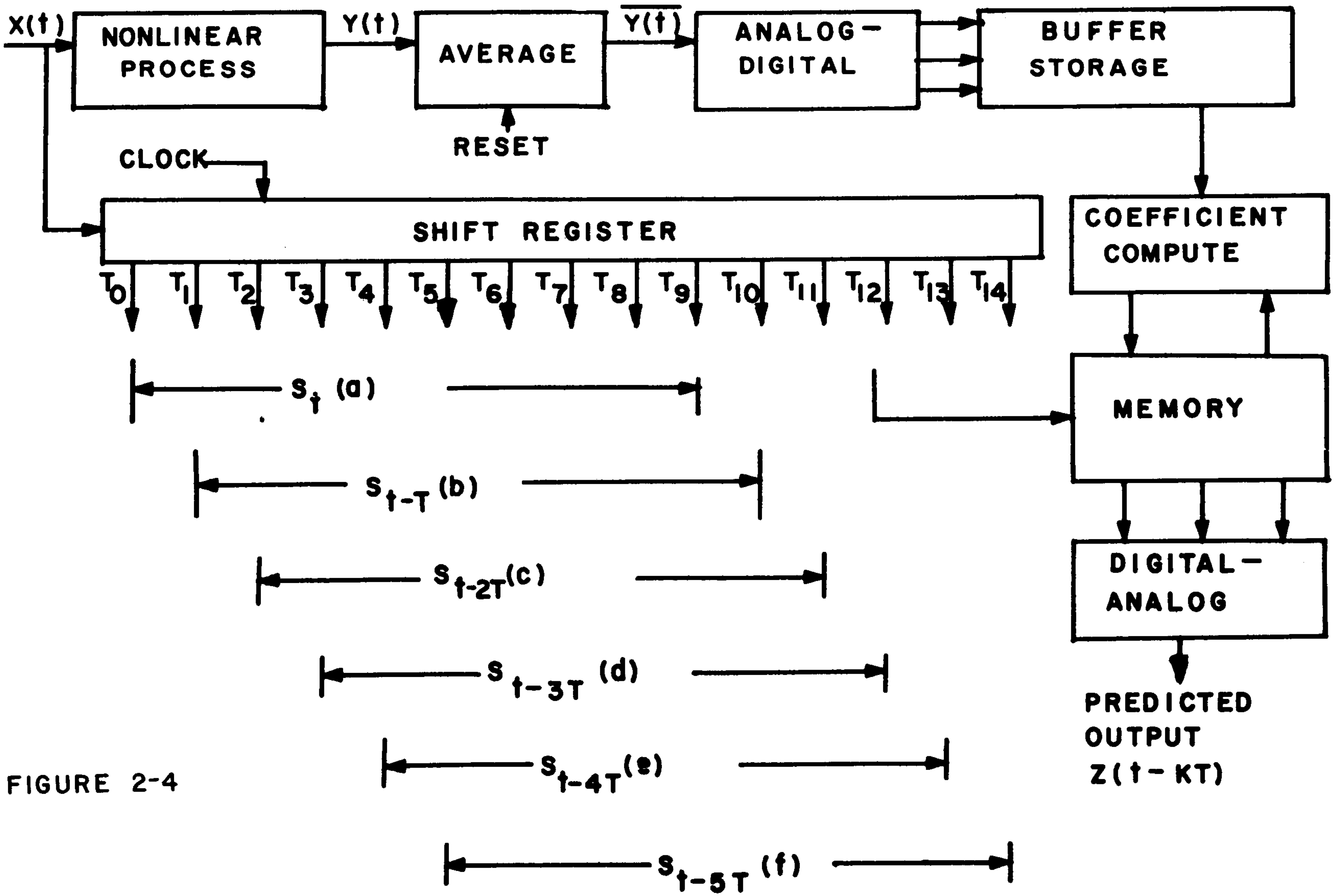

X(t)
NONLINEAR PROCESS
Y(t)
AVERAGE
RESET
$\overline{Y(t)}$
ANALOG-DIGITAL
BUFFER STORAGE
CLOCK
SHIFT REGISTER
COEFFICIENT COMPUTE
T_0
T_1
T_2
T_3
T_4
T_5
T_6
T_7
T_8
T_9
T_{10}
T_{11}
T_{12}
T_{13}
T_{14}
MEMORY
S_t (a)
S_{t-T} (b)
S_{t-2T} (c)
S_{t-3T} (d)
S_{t-4T} (e)
S_{t-5T} (f)
DIGITAL-ANALOG
PREDICTED OUTPUT
Z(t - KT)

FIGURE 2-4

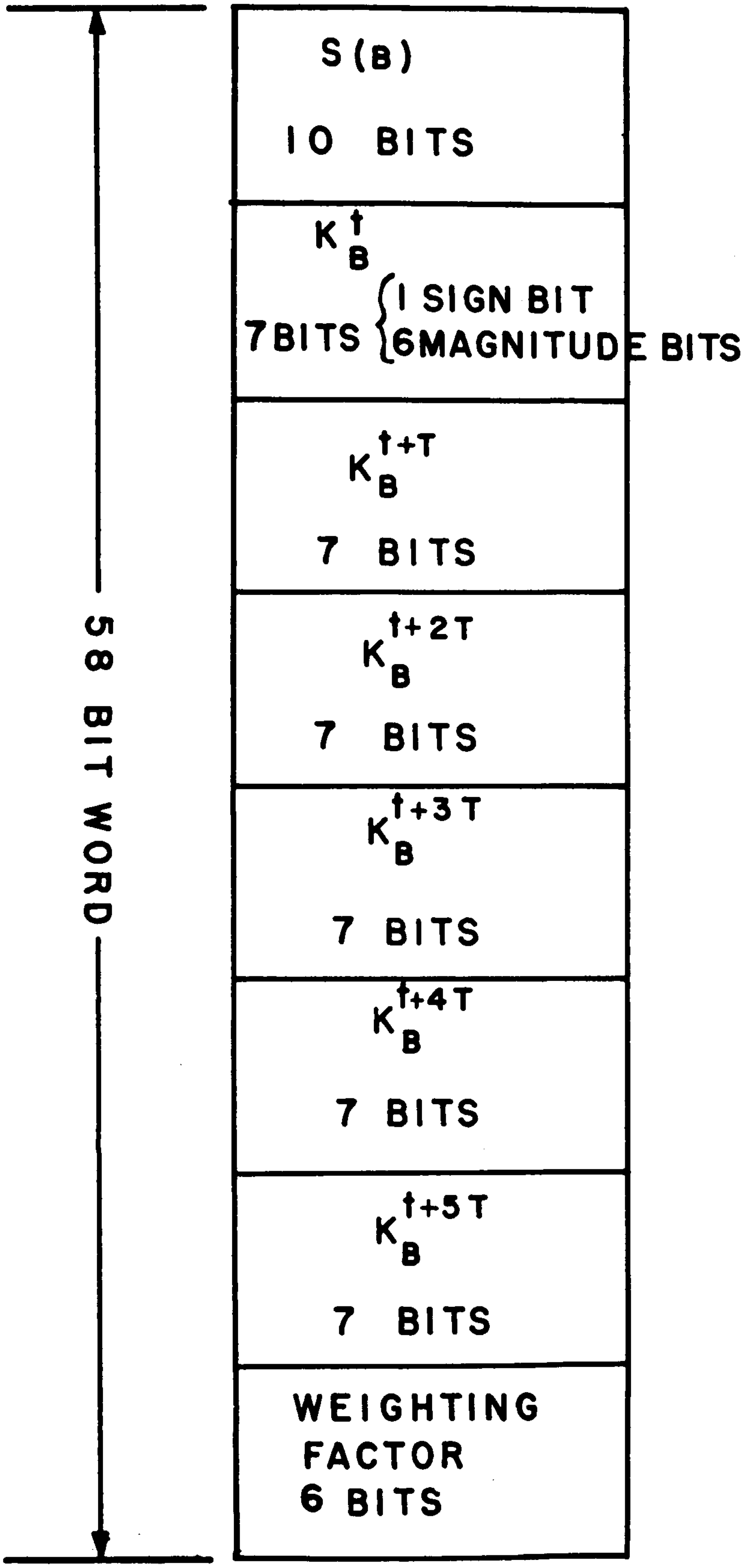
The problem considered is a deterministic one. Furthermore, a finite and fixed time interval is assumed. Under these restrictions, it appears that the brute force technique involving the calculation of partial sums is the best with regard to computation time, whereas that involving straightforward calculation requires the least memory capacity, since one need not store partial sums or minimum error functions. If the optimization interval is significantly longer than the memory of the input, however, dynamic programming is the only feasible method of the three.

2.11 Possible Model Structure

The model described above is easily implemented as a small special purpose computer. This section describes the general structure of such a machine. A more detailed description is given in Appendix IV.

A general configuration for the determination of the K_α's is shown in Figure 2-4. The input storage shift register shows the storage of the present sequence $S_t(a)$ and the storage of five past sequences $S_{t-T}(b)$, $S_{t-2T}(c)$, $S_{t-3T}(d)$, $S_{t-4T}(e)$, and $S_{t-5T}(f)$. In each memory location $S_t(a)$ there is stored a set of coefficients K_a, $K_a^{t + T}$, $K_a^{t + 2T}$, $K_a^{t + 3T}$, $K_a^{t + 4T}$, $K_a^{t + 5T}$ which are the model outputs $Z(t + T)$, $Z(t + 2T)$, $Z(t + 3T)$, $Z(t + 4T)$, and $Z(t + 5T)$, given the present input sequence $S_t(a)$. For each input sequence the present output is averaged, compared with the existing K_a^t and a new K_a^t stored in memory location $S(a)$. A new value is computed for $K_b^{t + T}$, $K_c^{t + 2T}$, $K_d^{t + 3T}$, $K_e^{t + 4T}$, $K_f^{t + 5T}$ and stored in the proper positions of the memory locations $S(b)$, $S(c)$, $S(d)$, $S(e)$, and $S(f)$ respectively.

FIGURE 2-5


58 BIT WORD
S(B)
10 BITS
K_B^t
7 BITS
1 SIGN BIT
6 MAGNITUDE BITS
K_B^{t+T}
7 BITS
K_B^{t+2T}
7 BITS
K_B^{t+3T}
7 BITS
K_B^{t+4T}
7 BITS
K_B^{t+5T}
7 BITS
WEIGHTING FACTOR
6 BITS

Since it is not known *a priori* exactly how many or which input sequences will occur, the detailed logical structure of the model cannot be designed without introducing a large number of redundant storage locations. For example, if 1024 switching circuits and 1024 memory locations were built (one for each of the possible 1024 sequences) and only 100 different sequences actually occurred, then there would be 924 switching circuits and 924 storage locations which were not required.

Therefore, it is advantageous to allow the input waveform to determine the specific structure of the model. This implies that the model be a serial access device. The concept employed is that of allowing each new sequence that appears to select the next available storage location and tag that location with the ten bits of the sequence. Assume then, that in each storage location it is desired that the present output K_β^t and the next five predicted outputs K_β^{t+T} to K_β^{t+5T} be stored. Seven bits of information will determine each coefficient to within 1%. (1 sign bit and 6 magnitude bits). For proper weighting of new information with the amount of data presently stored in each location, 6 weighting bits are assigned to each location. These 6 bits reflect the number of times that the sequence occurred in the immediate past, and enable the system to adapt to changing process parameters. Figure 2-5 shows a 58 bit word associated with each sequence. For a system with a dominant time constant of 100 milliseconds (about the lower limit expected) the length of an adequate input memory storage for determination K_β^t is approximately 700 milliseconds. Assuming that a ten tap memory is used, this implies that the minimum time between switching is 70 milliseconds. A memory access time of 1/10 of this value or 7 milliseconds is available. For this magnitude of access time, drum storage is sufficient.

Assuming then that drum storage is utilized, read in and read out of a storage location is determined by matching, in both the storage location and the input shift register, the 10 bits of the sequence. For read in purposes it will be necessary to have a buffer register to retain the information in the input shift register and the output average after the next sequence has been reached. The drum then seeks out locations S(a), S(b), S(c), S(d), S(e), and S(f) and stores the coefficients K_a^t, $K_b^{t + T}$, $K_c^{t + 2T}$, $K_d^{t + 4T}$, $K_f^{t + 5T}$ in their proper locations according to the word organization of Figure 2-5. On read out, the present sequence S(a) is sought and the predicted output is read out for $t = 0$ to $t = 5T$. In a system of this type there is no requirement for preassigned storage locations as the model is a serial access device.

A model structured as described was constructed for purposes of making experimental identification and prediction tests on processes simulated on an analog computer. The constructed model is fully described in Appendix IV.

2.12 Summary of Experimental Tests

The constructed model was used to experimentally verify the theory presented in this chapter. The tests are described in detail in Part VIII of Appendix IV.

In summary, four processes, ranging from linear to one with a hysteresis nonlinearity, were identified. The adaptive aspects of the model, that is the identification of a previously unseen process, were also tested. The results contained in Part VIII of Appendix IV are a good verification of the theory, and indicate that a practical model can be constructed and utilized.

2.13 Conclusions for Switched Two-level Input Processes

The type of identification presented in this chapter is essentially a table lookup procedure. It is a practical method for two level input signals. In this case, the memory requirements are not excessive. The model can be used for identification, and/or prediction, and these can be performed in an adaptive fashion. An identifier of this type could be used in a control system to control the identified process.

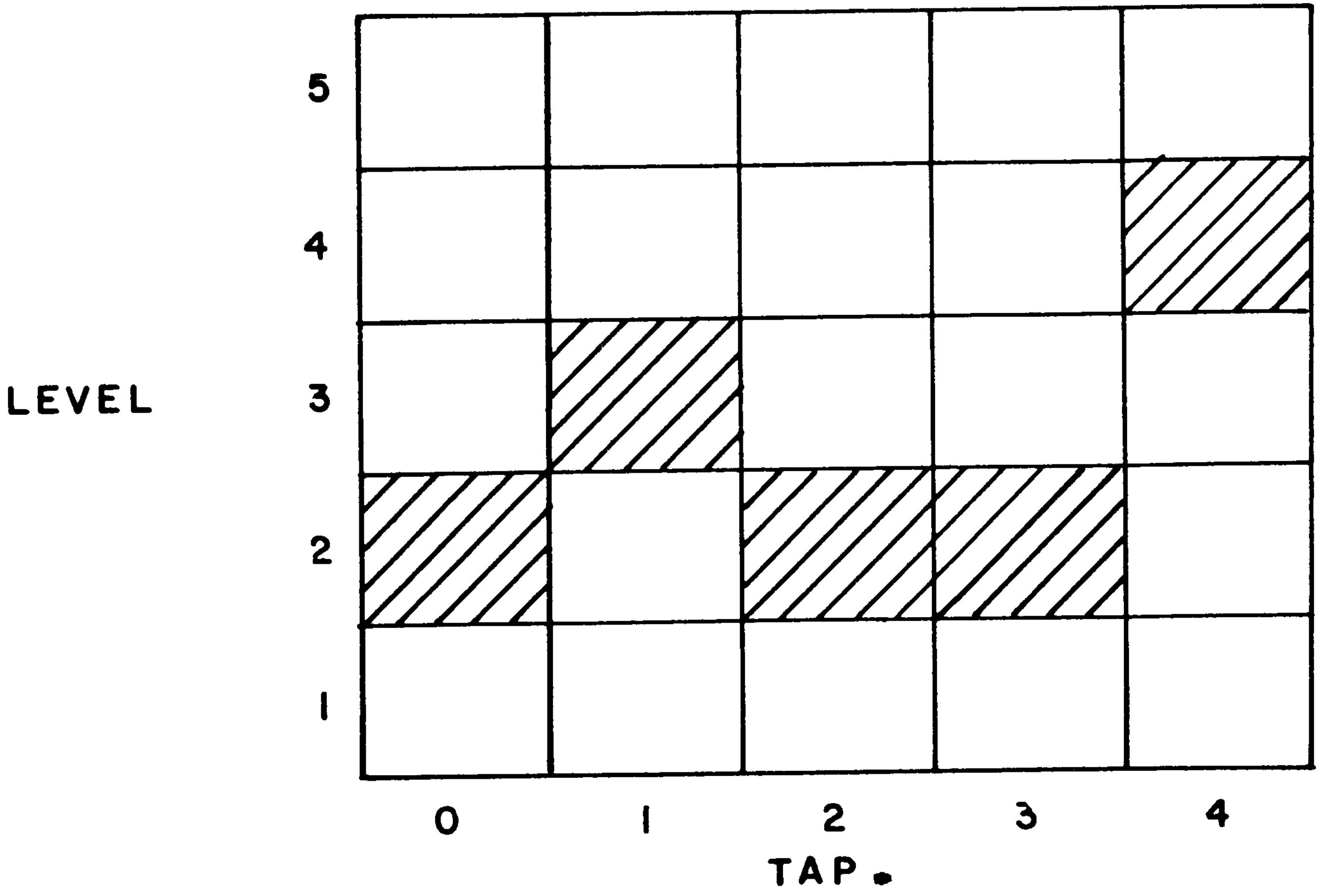


FIGURE 3-1

III General Processes

3.1 Introduction

For the class of processes with two level inputs, the type of identification presented in Chapter II is excellent, both from a practical viewpoint and from the viewpoint of minimum mean square error between process and model. It was shown that an adequate model can be obtained with a minimum of ten past input points. For a two level input, the number of possible input sequences or states using ten past input points is 2^{10} or 1024, which is not an unreasonable number. However, when an attempt is made to characterize processes which have inputs which are not two level, but are m level, the number of characterizing coefficients grows rapidly. For ten level input quantization and ten past input points, m = 10 and n = 10 yielding a figure of 10^{10} or 10 billion possible input states. This rather staggering figure closes the door on further investigation along the lines of Chapter II. Nevertheless, by stepping back and observing the general problem, there is a reasonable approach which this chapter explores. But much remains for future research.

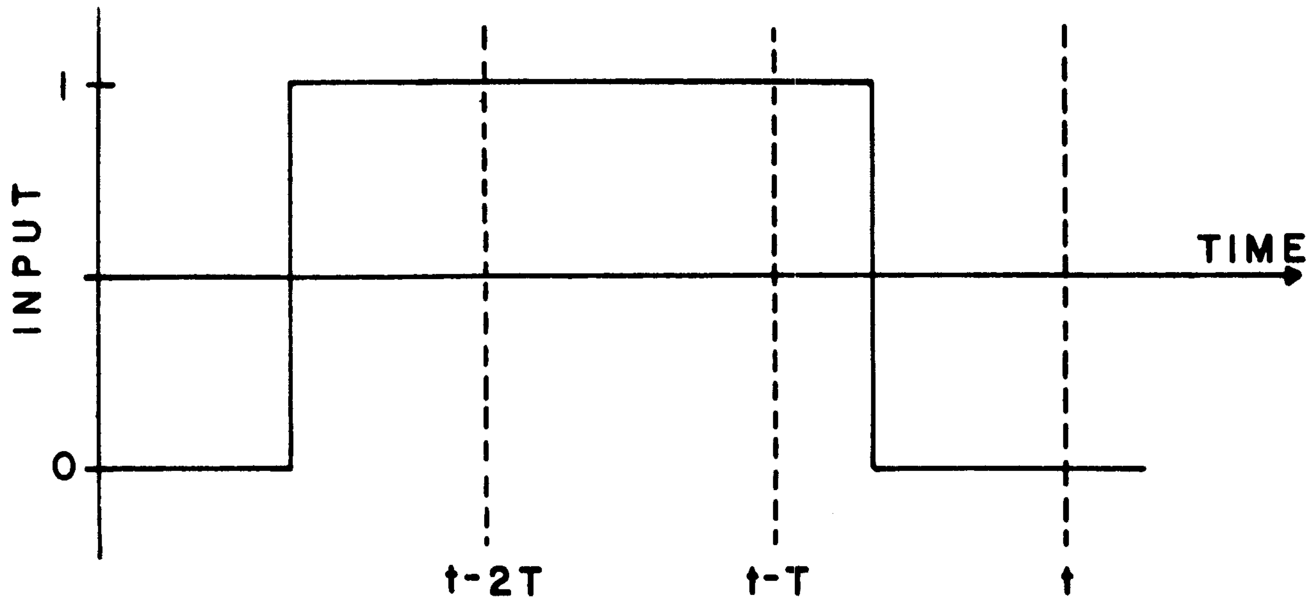
Basically, the problem under investigation is one of pattern recognition. For an input past obtained by quantizing into m levels, each of the n sample points, an m x n matrix is obtained. Shown in Figure 3-1 is a 5 x 5 matrix illustrating this point. There are 5^5 or 3125 possible states for this case. One of the possible 3125 possible input pasts is shown by the shaded boxes. The similarity between determining the proper model output and the identification of the proper pattern is immediately obvious if it is assumed that the output of the nonlinear process is quantized into p levels. Process identification is seen as the recognition (given a particular input pattern) of the proper character from among the alphabet of p characters.

Basically the work of the previous chapter on process identification was the recognition of a unique process outpt for a given input pattern. However, this required the storing of the entire list of possible input patterns for an m x n matrix. This list of m^n possible input patterns lies within reasonable storage facility for m = 2, but is unmanageable for the general case where m may be of the order of ten. A promising approach to the solution of the general problem exists in the techniques of pattern recognition.[9-13]

3.2 General Identification and Modeling with Probabilistic Decision Making

There are a finite number of samples of the input process. If bounds are known on the input and output, then one may consider a finite number of inputs and a finite number of outputs. For instance, if the input is sampled at n times and quantized into m different levels, then there are m^n different sampled-quantized representations of the input. If the input quantization levels are numbered 0 thru m-1 then one may represent any input sequence as an m- ary number $x^* = x_1 m^{n-1} + x_2 m^{n-2} + \ldots + x_{n-1} m + x_n$ where x_1 is the quantized value of $x(t)$; x_2, the quantized value of $x(t-T)$, etc. (The m- ary number x^* merely provides a convenient way of indexing the sampled-quantized inputs for reference, and there is no other significance to the value of x^*.)

For purposes of notation, and to tie this method in with those to follow, the table lookup method of Chapter II, generalized to m input



BINARY INPUT

$X^*=011$ (BINARY)

FIGURE 3-2

levels, may be viewed as follows. There are m^n different input representations x*. Let X be an m^n dimensional "vector" containing zeros except for a one in the x*th place, where x* is the m-ary number corresponding to the input representation of x. Thus for the input of Figure 3-2, x* = 011 in binary notation or x* = 3 in decimal notation. The vector X is $2^3 = 8$ dimensional, and X = (0,0,1,0,0,0,0,0).

Now define a matrix L = (1_{ij}) which is m^n by p, where p is the number of output levels. Let L be such that

$$\ell_{ij} = \begin{cases} 1 \text{ if input } x^* = i \text{ produces output } j \\ 0 \text{ otherwise} \end{cases}$$

Then the row vector D = XL will be a 1 by p matrix, whose i^{th} element will be zero or one according to whether the i^{th} output is or is not produced by inputs corresponding to X. There will be only one such element equal to "I" since, in the table lookup method, only one output is taken for each input. It is the one corresponding to the average output.

For instance, suppose that m = 2, n = 3 as in Figure 3-2, and that p = 2. This is a primitive example which will be carried throughout this chapter for purposes of illustration. Table 3-1 is assumed to describe the statistical properties of the plant. Notice that the table gives only probabilities, since error is assumed to be induced by sampling and quantization. (It is desired to decide only whether the output is positive or negative.) The matrix L, for this case is then

$$L = \begin{array}{cc|c} & & \text{Input} \\ \hline 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 2 \\ 0 & 1 & 3 \\ 1 & 0 & 4 \\ 0 & 1 & 5 \\ 0 & 1 & 6 \\ 0 & 1 & 7 \\ + & - & \\ \multicolumn{2}{c}{\text{Avg. Output Polarity}} & \end{array}$$

Input x*	Average Output	P(y, x*)	
		(y < 0) 0	(y > 0) 1
0	-0.5	1/8	0
1	-0.3	3/32	1/32
2	-0.1	5/64	3/64
3	0.4	1/32	3/32
4	-0.4	3/32	1/32
5	0.1	3/64	5/64
6	0.3	1/32	3/32
7	0.5	0	1/8

Table 3-1

Example

Assumed Process Properties

If the input x* = 3 is present, as in Figure 3-2, then

$$D = XL = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

indicating that $y > 0$ should be selected.

An alternate method is to make the matrix L such that $l_{ij} = P(i/j)$ probability that the i = x*th input is present if the j^{th} output were to appear. Then the k^{th} element of D - XL is a measure of the correlation between the input X and all other inputs which produce the k^{th} output, since

$$d_k = P(x^*/k)$$

This method and its extension (to be discussed later) is similar to that used by Highleyman for pattern recognition. The i^{th} row of L is an average value of X given the output y, and the input actually present is correlated with L. The decision made is to take the output k corresponding to the highest correlation d_k. More will be said about the relative merits of this method in a later section. In the example, the matrix L would be:

$$L = \begin{bmatrix} 1/4 & 0 \\ 3/16 & 1/16 \\ 5/32 & 3/32 \\ 1/16 & 3/16 \\ 3/16 & 1/16 \\ 3/32 & 5/32 \\ 1/16 & 3/16 \\ 0 & 1/4 \end{bmatrix}$$

where the probabilities of Table 3-1 have been divided by P(y) = 1/2 to convert from joint probabilities to conditional probabilities.

The vector D = XL for x* = 3 is then

$$[D] = [1/16 \qquad 3/16]$$

so that $y > 0$ is the decision made by the model.

3.3 Reduction of Information

Of course, the trouble with the table lookup method is that m^n is usually far too many numbers or probabilities to remember. Thus the amount of information in the inputs must be condensed or reduced in such a manner as to make realization of the model feasible, and at the same time, to retain reasonable accuracy. Of course, it is not possible to name any specific method of doing this which will work for all non-linear systems, since the mapping from the x(t) to y is arbitrary. For this reason, several such methods are postulated, and the various advantages and disadvantages discussed.

Instead of measuring the entire input pattern x*, we measure certain "features" of the pattern, say $\alpha_1, \alpha_2, \ldots, \alpha_v$; where each α_i is a variable to which the values 0, 1, ..., S_i-1 may be assigned, depending upon the x* which is present. S_i is analogous to m of the previous case. Now let A be a row vector whose first S_1 elements denote the state of α_1. All elements of the first S_1 elements being 0 except the $\alpha_1 + 1^{st}$, which is one. Likewise for the next S_2 elements and α_2, etc. up to α_v and S_v. Thus $A = (s_i)$ has $S = \sum_{i=1}^{v} S_i$ elements of which v are one and the rest zero.

3.4 Binary Learning Matrix

Now, as before, let $L = (\ell_{ij})$ be a S x p matrix (p = number of output levels), whose elements are

$$\ell_{ij} = \begin{cases} 1 & \text{if the jth output was produced for an input with } a_i = 1 \text{ previously.} \\ 0 & \text{otherwise} \end{cases}$$

Thus L is updated after each input over a certain learning period to conform to the above criterion. Now the decision is made by letting D = AL, and choosing the output "k" corresponding to the largest d_k. This is similar to the Browning-Bledsoe method for pattern recognition.[9] The learning period must be limited due to an effect they refer to as "saturation". The maximum value of any element of the vector D is S. This will occur when a representation x* appears which has appeared previously, producing the same output. It will also occur if each of the α's in the representation A occurred previously (but not necessarily simultaneously) producing a certain output. Depending on the features α measured, of course, this latter may be a fairly likely occurrence if enough inputs have been observed. The trouble comes in when this saturation is observed for the same set of α's but different outputs. Then the matrix D may have two or more elements equal to the maximum value v, and a "tie" results. One does not know which output to choose. (For lack of a better criterion, the average is taken.) The advantage of this method is that only binary numbers need be stored in the matrix L, which is the largest factor in determining storage requirements.

3.5 Probabilistic Learning Matrix

If more storage space is available, then it may be possible to eliminate the "saturation" problem by using conditional probabilities for the matrix L.

$\ell_{ij} = P(a_i/j)$ the probability that the i^{th} element of the input matrix A is a_i, given the jth output present.

In this method, the output is found as before. That is, the matrix product D = AL is taken, and the jth output is taken as the model output if d_j is the largest element of D. The matrix L is determined by analyzing the input-output operating record. The frequency with which each combination (a_i, y_j) occur, is measured. These frequencies will converge to the corresponding probabilities for long operating records. The conditional probability $P(a_i/j)$ is determined by dividing the number of times (a_i, y_j) has occurred by the number of times the output y_i has occurred. The multiplication D = AL is alternately written as

$$d_j = \sum_{i=1}^{N} a_i \; P(a_i/y_j); \; j = 1, \dots, p$$

In this form, it is easily seen that (since the a_i are binary) d_j represents the correlation between the "input" $A = (a_i)$ and the average value of A given that the output is y_j. Thus the procedure amounts to correlating the given A with an "average input" producing y_j. The element d_j is the inner product of the vector A and the "average vector" producing the jth output, which is the jth column vector of L.

Considering an example, suppose only two "features" of the input in Figure 3-2 are taken, namely the pair of samples at t and t-T together (call it α_1), and the second feature, the sample at t-2T, designated α_2.

α_1 has four possible states, given by $\alpha_1 = 2x(t) + x(t-T)$, and α_2 has two possible states given by $\alpha_2 = x(t-2T)$. For Figure 3-2 $\alpha_1 = 1$, $\alpha_2 = 1$. Thus $A = (0, 1, 0, 0, 0, 1)$, where the first four elements denote the state of α_1 and the last two denote the state of α_2.

Consider first the method wherein ℓ_{ij} is binary. Suppose inputs $x^* = 0, 1, 2, 5, 2, 5$ have occurred previously, with corresponding outputs 0, 0, 0, 1, 1, 1 respectively. Then the corresponding features (α_1, α_2) were (0, 0); (0, 1); (1, 0); (2, 1); (1, 0); (2, 1), and

$$L = \left[\begin{array}{cc} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \\ \hline 1 & 1 \\ 1 & 1 \end{array}\right] \begin{array}{c} \\ \\ \alpha_1 \\ \\ \\ \alpha_2 \\ \end{array}$$

and $D = AX = \begin{bmatrix} 2 & 2 \end{bmatrix}$.

Note that the decision cannot be made as saturation has already occurred. Using the method wherein $\ell_{ij} = P(i/j)$

$$L = \left[\begin{array}{cc} 7/16 & 1/16 \\ 7/32 & 9/32 \\ 9/32 & 7/32 \\ 1/16 & 7/16 \\ \hline 21/32 & 11/32 \\ 11/32 & 21/32 \end{array}\right] \begin{array}{c} \\ \\ \alpha_1 \\ \\ \\ \alpha_2 \\ \end{array}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$D = \begin{bmatrix} 9/16 & 15/16 \end{bmatrix}$$

and the decision is $y > 0$.

3.6 k-Tuples as Features

If the only a priori knowledge of the process is the settling time, and the bounds on the input and output, then there is no basis for choosing the features to be tested. In that event, one may follow the method of Browning and Bledsoe for pattern recognition. This consists in taking the features as k-tuples (combinations of k elements) of values from the original pattern or input representation x^*. Thus, for α_1 we may take the values of the quantized input at k sample points; for α_2, the values at k different sample points, etc. Thus each of the α's could be represented as an m-ary integer, $0 \leq \alpha_i \leq m^k-1$, in the same manner as x^*. If there are n samples of an input quantized into m levels, then the number of possible k-tuples is $\binom{n}{k} = \frac{n!}{(n-k)!\ k!}$. If the output is quantized into p levels, then the matrix L would be $m^k \binom{n}{k} \times p$, if all k-tuples are used; since there are m^k different possibilities for each k-tuple. If only "disjoint" k-tuples are used (i.e. no two k-tuples contain the same input sample) then L is $m^k\ (n/k) \times p$ (assuming k divides n). The table lookup method appears as a special case where $k = n$.

Table 3-2 gives the number of elements in L as a function of k for the case where $m = n = p = 10$. It is evident that considerable savings can be made in the storage requirements by using these features rather than the table lookup method ($k = 10$).

3.7 Decision Function Approach

The problem with the foregoing methods is precisely that they are specialized pattern recognition techniques. For although the process identification problem can be made to look like a pattern recognition problem, there are differences which cannot be ignored:

1. Error criteria may be different. If a pattern is mistaken for another there is usually no way of predicting whether or not that error is more serious than any other error which might have been made. On the other hand, for process identification, the greater the difference between model and process outputs, the more serious is the error.

Length of Sequences k	Memory Size
1	10^3
2	4.5×10^4
3	1.2×10^6
4	2.1×10^7
5	2.52×10^8
6	2.1×10^9
7	1.2×10^{10}
8	4.5×10^{10}
9	10^{11}
10	10^{11}

Table 3-2

Memory Sizes for Ten Samples and Ten Quantization Levels (taking all k-tuples)

2. The concept of an "average representation of a certain pattern" is a useful one, easily visualized, especially for recognition of typewritten letters and numerals. The corresponding concept of an "average input" which produces a certain output is far more nebulous. It is possible that two widely different (appearing) input functions could produce the same output.

The disadvantages of the method can be demonstrated via a simple counter-example. Suppose x is a single sample of a Gaussian input function with zero mean and unity variance. Suppose that the system is linear, and that the output y has unity variance, and that the correlation between x and y is ρ. We wish to make a decision about y based on the criterion of maximizing p(x/y):

$$p(x/y) = \frac{.\,1}{2\pi\,(1-p^2)^{1/2}}\; e^{-\frac{(x-py)^2}{2\,(1-p^2)}}$$

This is maximum when $z = \frac{x}{p}$. However, using a least MSE criterion, we should take y equal to the average value of y, given x:

$$z = \int_{-\infty}^{\infty} y\; p(y/x)\; dy$$

or

$$z = \rho\, x$$

The above arguments do not invalidate the methods discussed up to this point, since they can still be made to yield reasonable results in many cases. Instead, these disadvantages are listed in order to motivate what follows.

In the general classification problem, to which process identification, as formulated above, belongs; one applies decision theory in order to obtain a solution. One defines a loss function $\omega \left[y-z\ (x^*)\right]$ where y is the actual output and $z(x^*)$ is the model output (a function of the model input x^*) to be determined. The choice of ω depends upon the circumstances, but is usually some function of $|y - z|$ (the error), and which is zero when $|y - z|$ is zero, and is monotonic increasing in $|y - z|$. The problem is solved if one can choose $z(x^*)$ to minimize the expected value of ω:

$$E\ [\omega] \approx \sum_{i=1}^{p} \omega(y_i z)\ P(y_i/x^*)$$

Call the "best model output" z' then:

$$z' = \min_{z} \sum_{i=1}^{p} \omega(y_i - z)\ P(y_i/x^*)$$

Now if $P(y_i/x^*)$ is known then it is possible to find z'. For instance, if $\omega(x) = x^2$, then the model output should be chosen as

$$z = E\ [y] = \sum_{i=1}^{p} y_i\ P(y_i/x^*)$$

The problem is, of course, that the distribution $P(y_i/x^*)$ is not known, and is of too high an order to measure conveniently. What can be measured are the lower order distributions corresponding to "features" of the "pattern" x^*. These are $P_j(y_i, \alpha_n)$. The question arises, is there a way to approximate P from the P_j's?

Any high order distribution which is formed from the P_j and which converges to the P_j when summed over the appropriate variables (features not included in P_j) can be considered an approximation to P. Lewis has shown that the one which has maximum entropy is the one formed from the product of the P_j, if, indeed, it is such an approximation.[15]

Consider the probability distributions $P_i(\alpha_j/y_i)$. The α_j are taken to be disjoint q-tuples of samples of the input. The product:

$$P(y_i) \prod_{j=1}^{N} P_i(\alpha_j/y_i) = Q\ (\alpha_1, \ldots, \alpha_N, y_i)$$

is an approximation to $P(\alpha_1, \ldots, \alpha_N, y_i)$ since, summing over all α_j, $j \neq k$ produces $P(\alpha_k, y_i)$. (The α's are disjoint.) Hence, it is the approximation which has maximum entropy.

Choosing the approximation which has maximum entropy essentially means choosing that distribution with maximum "randomness" or least bias. It yields a distribution wherein the features α_j are statistically independent, _when the output y_i is given_. Although the α's are probably not independent in this sense, it is (by hypothesis) not possible to measure their dependence and still keep the storage requirement reasonable. Since their bias cannot be measured, it is not known which way they are biased. That is, it is not known whether $P(\alpha_j, \alpha_k/y_i)$ is greater or less than $P(\alpha_j/y_i)\ P(\alpha_k/y_i)$. Therefore, the most reasonable thing to do is to assume equality.

Using the approximation Q, the techniques of decision theory may be applied directly. The result will not actually minimize the expected loss, since Q is only an approximation. Instead, it will minimize a best guess of the expected loss, based on the information available.

For a minimum MSE criterion, the output to be taken will be:

$$y_m = \frac{\sum_{i=1}^{p} y_i\ Q\ (\alpha_1, \ldots, \alpha_N, y_i)}{\sum_{i=1}^{p} Q\ (\alpha_1, \ldots, \alpha_N, y_i)}$$

where the quotient:

$$\frac{Q\ (\alpha_1, \ldots, \alpha_N, y_i)}{\sum_{i=1}^{p} Q\ (\alpha_1, \ldots, \alpha_N, y_i)}$$

is the approximation to $P(y_i/\alpha_1, \ldots, \alpha_n)$ derived from Q.

As an example, suppose the system described by Table 3-1 is to be identified using this method, with the features α_1, α_2 as the (t, t-T) pair and the t-2T sample, respectively. Suppose the input is x* = 3 as in Figure 3-2. Then $(\alpha_1, \alpha_2) = (1,1)$, and the product approximation Q is given by

$$Q_i = P(\alpha_1/y_i)\ P(\alpha_2/y_i)\ P(y_i)$$

Note $P(y_i) = 1/2$, for all i. Then

$$Q_o = 0.376$$

$$Q_1 = .0920$$

and the approximations to $P(y_i/\alpha_1, \alpha_2)$ are

$$P(0/\alpha_1, \alpha_2) = .29$$

$$P(1/\alpha_1, \alpha_2) = .71$$

Now the actual values are 0.25 and 0.75, respectively, so that our error should be small.

For instance, if $y_i = \pm\ 0.5$ is chosen, then

$$y_m = \sum_{i=o}^{1} y_i\ P(y_i/\alpha_1, \alpha_2) = 0.21$$

The error depends on the values chosen for the quantized levels, but in any case $y_m > 0$.

3.8 Identification Time

Considerable improvement over the method of Chapter II may be possible by taking features of the input "pattern". For instance, suppose pairs of input samples are taken and the joint frequencies of output and input features are measured, using ten levels of quantization. Then there are 1000 possibilities for each combination of output and input feature. The frequency measured can be written as

$$P_j = \frac{1}{N} \sum_{i=1}^{N} f_i \, (\alpha_i, y_i) \qquad j = 1, \ldots, 1000$$

where $$f_i = \begin{cases} 1 \text{ when } (\alpha_i, y_i) \text{ is the jth such combination} \\ 0 \text{ otherwise} \end{cases}$$

where N successive combinations are observed.

Assuming (for simplicity) that the input-output combinations (α_i, y_i) are independent, then the expected value of P_j is the corresponding joint probability P_j. The variance of P is $\frac{1}{N} (P_j - P_j^2)$.

The normalized variance of P is thus

$$\frac{1}{N} \quad \frac{1 - P_j}{P_j}$$

which represents the percent MSE in the measurement of P. The average value of this is

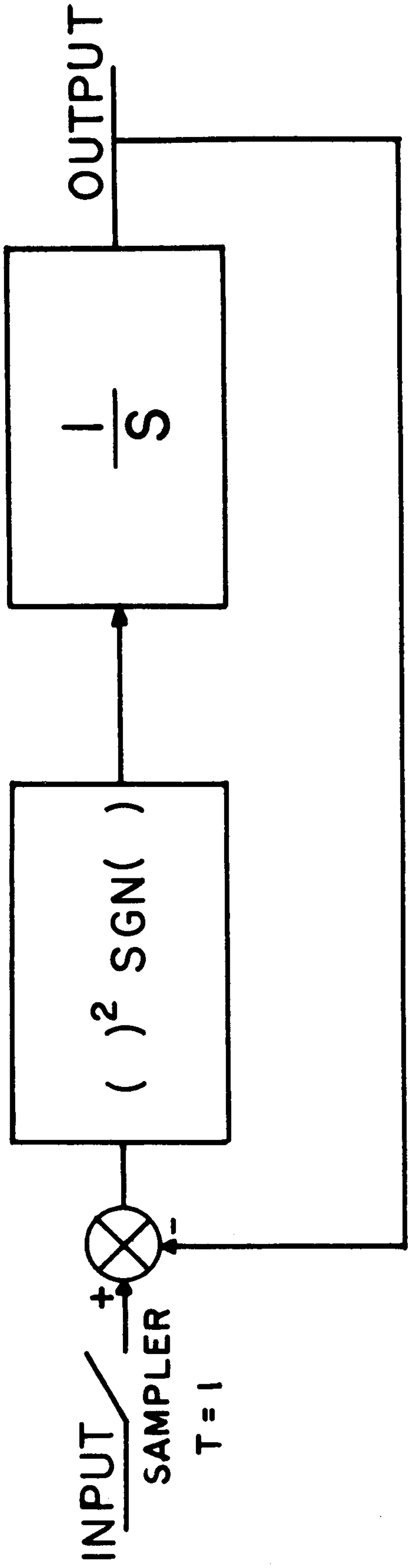


FIGURE 3-3

$$\frac{1}{N} \sum_{j=1}^{1000} (1-P_j) \approx \frac{1000}{N}$$

Thus, if it is desired to bring the average RMS error in measurement of the probabilities down to within 10%, one needs at least 10^5 measurements or roughly 10^4 settling times. Granted, the errors may in some cases be greater than this, but they will be smaller for the more likely input features. Likewise, for 1% RMS error, one requires 10^6 settling times.

The above figures are certainly not an accurate analysis of the identification times required. They are intended merely to give an idea of the magnitudes involved. It is apparent from these figures that the identification times involved are far too long to take care of most time varying systems, except possibly for the case of two level inputs.

3.9 Computer Results

The binary method was used to identify the non-linear sampled-data system of Figure 3-3, with k = 2 and m = n = 10. The input used was a sample function from a Gaussian random process with zero-mean, unity standard deviation and exponential autocorrelation function. The correlation between successive samples was roughly 0.7, and the sampling interval of the model coincided with that of the system, which was one second. Since the input is sampled and passed through a hold circuit, the output of the hold circuit is a sequence of step functions of varying height.

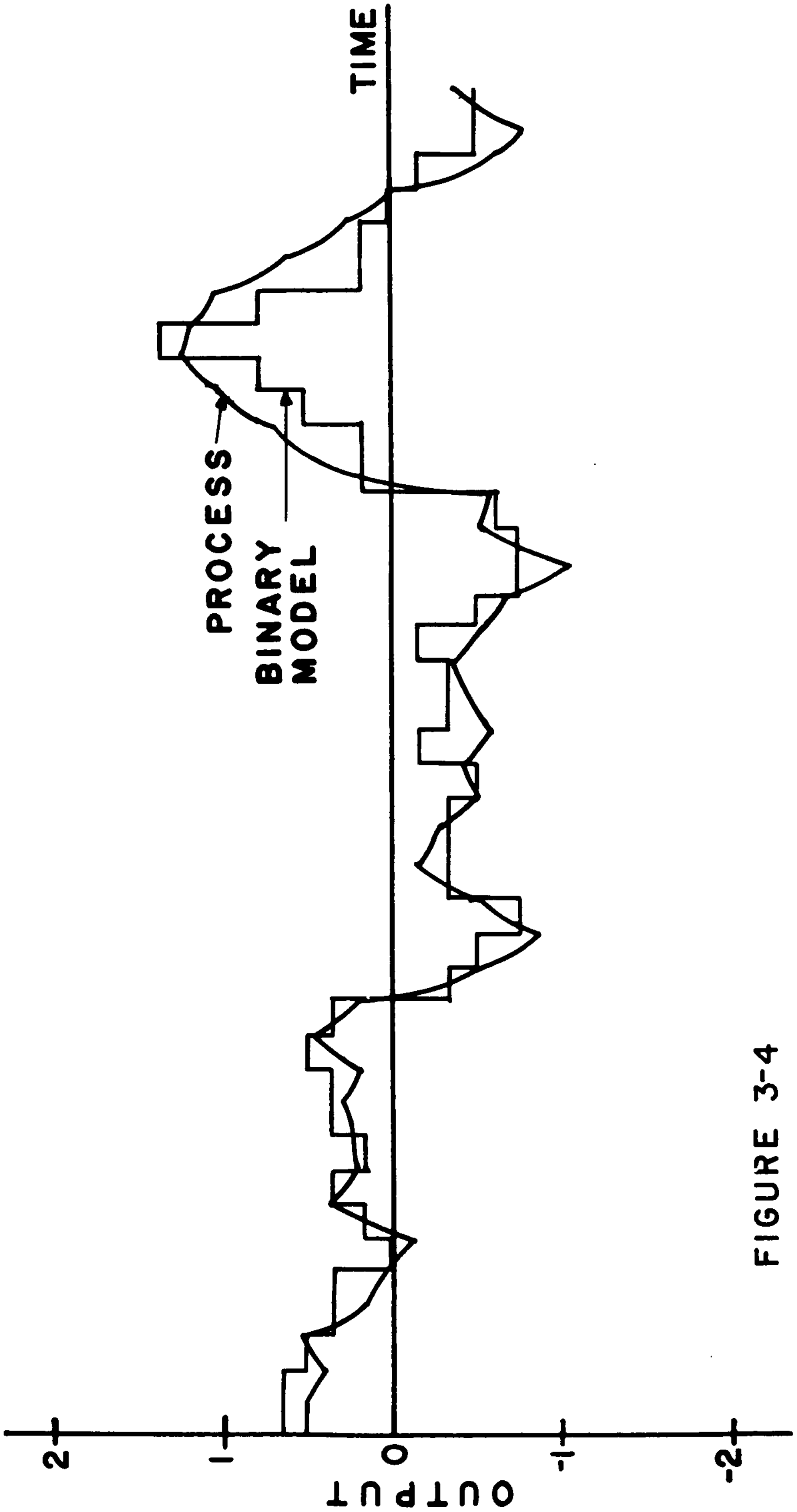


FIGURE 3-4

The model used is a "learning matrix" of binary elements. The matrix has 45,000 elements; that is, 45 different pairs of input sample times, and 100 different "states" for each pair, for 10 outputs. The ten quantization levels for the input were chosen on the basis of minimum "distortion" in the mean square sense.[17] That is, the quantization levels are chosen so as to minimize the mean square quantization noise.

An IBM 704 computer was used to simulate both the system and the model. For each input, the average output was computed and compared to the output predicted by the model.

An input record of 2,000 samples was used. At the end of this time the "saturation" effect was noticeable; that is, a large number of ones are entered in the learning matrices, so that "ties" involving two or more outputs having the maximum score (in this case, 45) occur quite frequently.

Figure 3-4 shows a typical section of the system output and the corresponding model output after about 1900 input samples had been taken. Figure 3-5 shows the frequency of proper identification vs. time, obtained from this data. The dashed line is the frequency of "close" identification (plus or minus one quantization level). Of course, one should not expect these curves to be smooth, since the input to the system was random. It is evident, however, that the results are as would be expected. The frequency of identification increases as the model "learns" more about the system, then decreases due to saturation of the learning matrices. Notice that, for an error of one level or less, the "learning" curve continues to increase, since "ties" are most likely

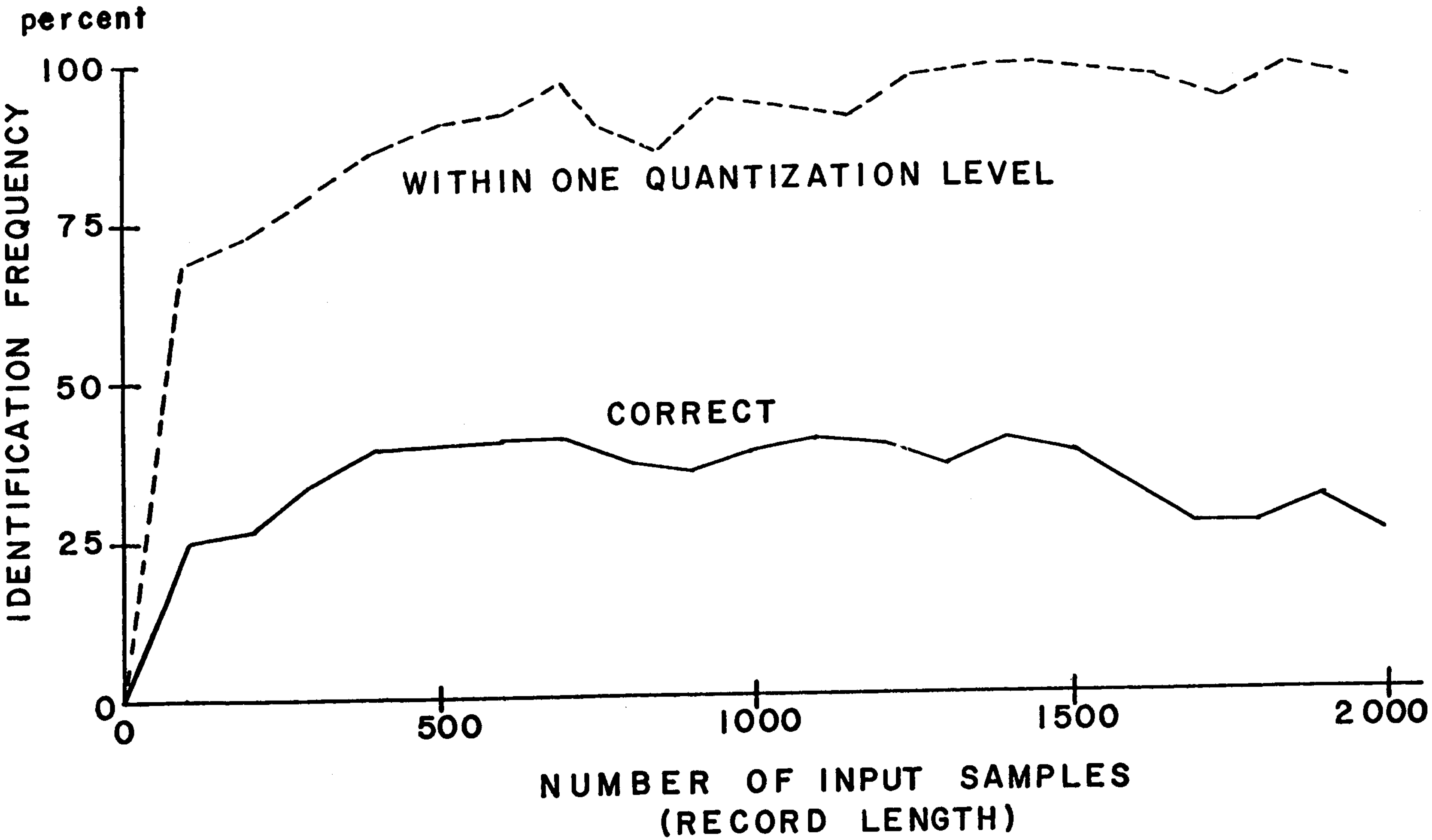

percent
IDENTIFICATION FREQUENCY
100
75
50
25
0
0
500
1000
1500
2000
WITHIN ONE QUANTIZATION LEVEL
CORRECT
NUMBER OF INPUT SAMPLES
(RECORD LENGTH)

FIGURE 3-5

to occur between adjacent levels. If the data had been carried further, this too would have started to decrease.

For binary learning matrices, two points are evident:

(1) Quantization must be considerably finer than the accuracy required.

(2) The "learning" process must be terminated at some point if accuracy is to be maintained.

This same procedure was used on the non-linear system with two level input shown in Fig. 3-6. Only nine pairs of input samples were used. However instead of binary elements in the learning matrix, probabilities were used, as in Section 3.5. Figure 3-7 is a typical plot of the system output and the model output. The data was obtained by simulating the model on an IBM 1410 computer. Identification is apparently quite good using this model.

3.10 Conclusions for Identification of General Processes

Although much remains to be studied, the results obtained in this study show the following:

1. Table lookup methods are quite feasible for nonlinear process with a two level input. Beyond two level inputs, this method becomes impractical.

2. Binary learning matrices work quite well for nonlinear processes with two level inputs, reducing the memory requirements below that required for a table lookup method. For processes with a multilevel input, this method begins to fail due to "learning matrix saturation".

3. Learning matrices which store conditional probabilities overcome the problem of "learning matrix saturation" but increase

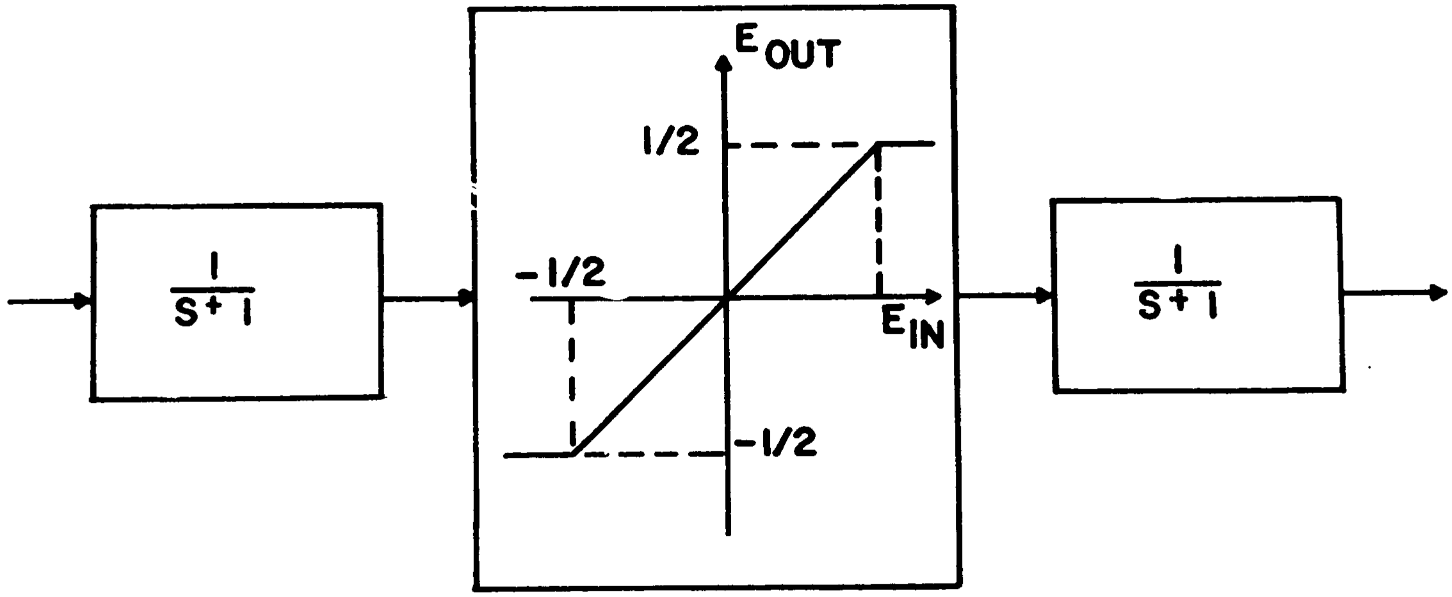


FIGURE 3-6

the memory requirements over that required for binary learning matrices. The memory requirements for this method, for a multi-level input, are still within the practical region, whereas for a table lookup method the memory requirements are astronomical.

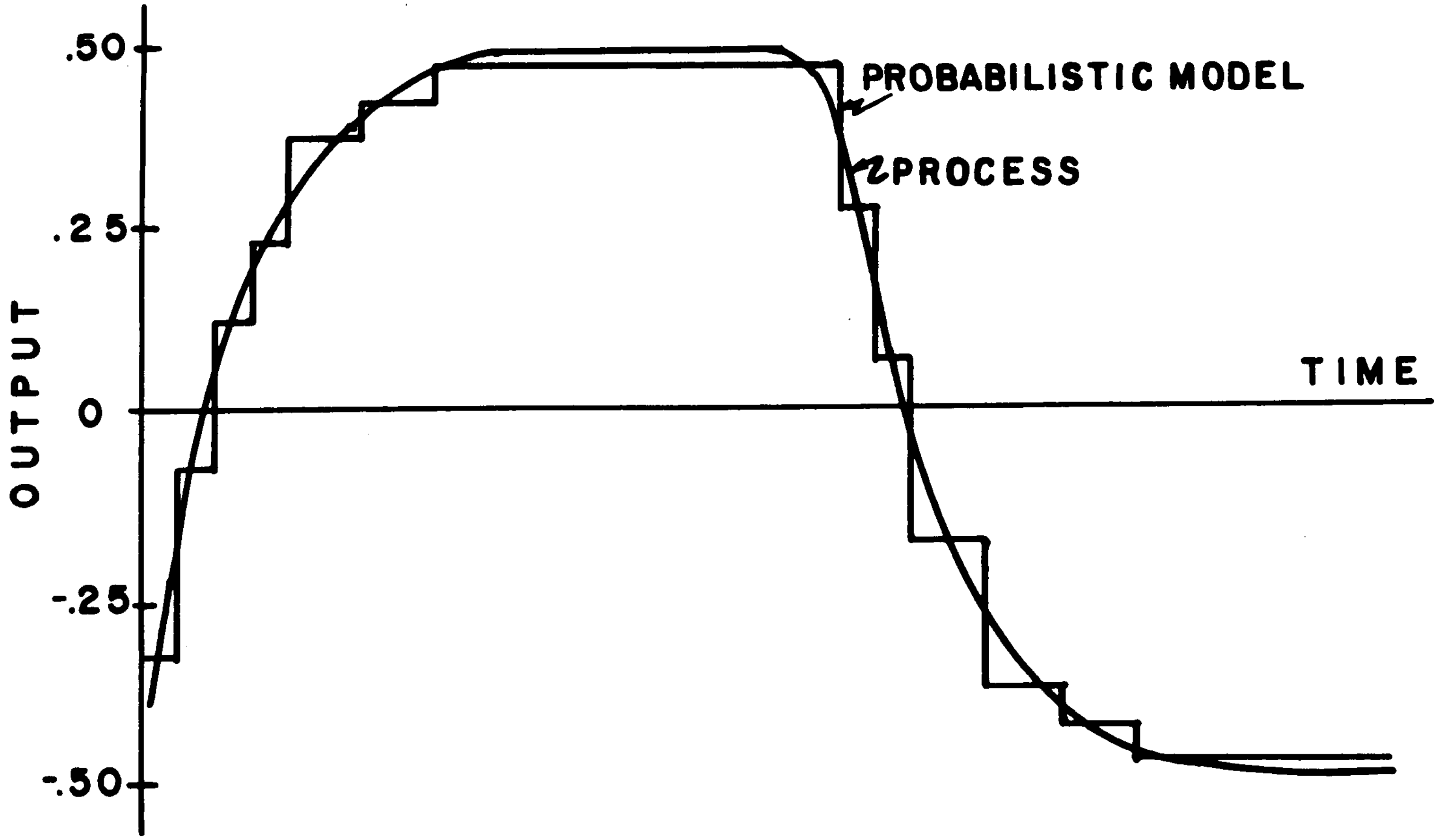


FIGURE 3-7

REFERENCES

1. Mishkin, E. and Braun, L., Jr., "Adaptive Control Systems" (book), McGraw-Hill Book Co., New York, 1961, Chapter 3.

2. Cooper, G. R. and Lindenlaub, J. C., "Limits on the Identification Time for Linear Systems", Purdue University Tech. Rept. No. 2, Project 8225, 1961.

3. Coales, Jr. F., and Noton, A. R. N., "An On-Off Servomechanism with Predicted Change-Over", Proc. Inst. Elect. Engrs., vol 103, pt. B, August 1955, pp. 449-62.

4. Chestnut, H., Sollecito, W. E., and Troutman, P. H., "Predictive Control System Application", A.I.E.E., pt. 2, Appl. and Industry, July 1961, pp. 128-39.

5. Goodman, T. P. and Reswick, J. B., "Determination of System Characteristics from Normal Operating Records", Trans. ASME, vol 78, Feb. 1956, pp. 259-71.

6. Goodman, T. P., "Theory and Application of a Tapered Electronic Delay-line Synthesizer", Proc. First IFAC Congress, vol III, pp. 102-108.

7. Bose, A. G., "A Theory of Nonlinear Systems", M.I.T. Res. Lab. of Electronics Tech. Rept. 309, 1956.

8. Feller, W., "Probability Theory and Its Applications", (book), John Wiley and Sons, 1950, pp. 91-4, 209-11.

9. Bledsoe, W. W. and Browning, I., "Pattern Recognition and Reading by Machine", 1959 Proceedings of the Eastern Joint Computer Conference, Sandia Corporation Reprint, SCR-132, March 1960.

10. Bledsoe, W. W., "Further Results on the N-tuple Pattern Recognition Method", IRE-PGEC, March 1961, p. 96.

11. Bledsoe, W. W., and Bisson, C. L., "Improved Memory Matrices for the N-tuple Pattern Recognition Method", IRE-PGEC, June 1962, pp. 414-15.

12. Highleyman, W. H., "Further Comments on the N-tuple Pattern Recognition Method", IRE-PGEC, March 1961, p.97.

13. Steck, G. P. "Stochastic Model for the Browning-Bledsoe Pattern Recognition Scheme, IRE-PGEC, April 1962, pp. 274-82.

14. Highleyman, W. H., "An Analog Method for Character Recognition", IRE Trans. on Electronic Computers, Sept. 1961, pp. 502-12.

15. Lewis, P. M. "Approximating Probability Distributions to Reduce Storage Requirements", Information and Control, vol 2, No. 3, 1959, pp. 214-25.

16. Davenport, W. B. and Root, W. L., "An Introduction to the Theory of Random Signals and Noise, (book), McGraw-Hill Book Co., 1958, pp. 77-9.

17. Max, J. "Quantizing for Minimum Distortion", IRE Trans PGIT, March 1960, pp. 7-12.

APPENDIX I

MSE Due to Coefficient Inaccuracy - Two Level Case

The coefficients (K_α) are determined by taking the time average of the output during the time that the input sequence is $S(\alpha)$. The coefficients determined in this manner will be very nearly the correct values, as given by Eq. 2-8, provided the averaging is done over a sufficiently long period of time. Since, however, averaging must be done over a finite length of time, the coefficients computed in this manner will be in error. This will cause the mean square error of the model output (E) to be greater than the ideal, or minimum value (E_{min}). This additional error depends on the coefficients computed, which are randomly distributed about K_α. An estimate of the additional MSE caused by inaccuracy of the coefficients can be obtained by averaging this error over all possible values of the computed coefficient. This average additional MSE is always less than E_{min}. Furthermore, if each sequence occurs at least M times in the operating record, then the average additional MSE is at most $(1/M)\ E_{min}$. The above statements are proven as follows:

Denote the "true" coefficients as given by Eq. 2-8 by K_α. Denote the coefficients computed by time averaging by K'_α. Let $< \; >$ denote statistical average over K'_α, which is a random variable. The minimum MSE is found by combining Eqs. 2-8 and 2-5.

$$E_{min} = \overline{y(t + T)^2} - \sum_\alpha K_\alpha^2 \ \overline{S(\alpha)} \qquad \text{I-1}$$

Notice that, since $\overline{S(\alpha)}$ is merely the probability that the sequence $S(\alpha)$ occurs, the last term of Eq. I-1 is the mean square value of the model output.

The MSE, if the coefficients K'_α are used, is given by using K'_α in Eq. 2-5. Subtracting Eq. I-1,

$$E - E_{min} = \sum_\alpha (K_\alpha^2 - 2\, K'_\alpha\, K_\alpha + K'^2_\alpha)\, \overline{S(\alpha)}$$

Averaging over all K'_α,

$$\langle E - E_{min} \rangle = \sum_\alpha (K_\alpha^2 - 2 \; \langle K'_\alpha \rangle \; K_\alpha + \langle K'^2_\alpha \rangle)\, \overline{S(\alpha)} \quad \text{I-2}$$

But, for every α,

$$\langle K'_\alpha \rangle = K_\alpha \quad \text{I-3}$$

and

$$\sum_\alpha \langle K'^2_\alpha \rangle \; \overline{S(\alpha)} \leq \overline{y(t + T)^2} \quad \text{I-4}$$

The left hand side of Eq. I-4 is a statistical average of the square of the time average of the output over some finite interval. This may never exceed the mean square output, which is the right hand side.

Combining Eq. I-2, I-3, and I-4,

$$\langle E \rangle \; - E_{min} \leq \overline{y(t + T)^2} - \sum_\alpha K_\alpha^2 \; \overline{S(\alpha)}$$

or,

$$\langle E \rangle \leq 2\, E_{min}$$

Thus, the average effect of errors due to a finite input record is, at worst, to double the MSE.

If the computations of the K_α are based on an input record containing each sequence at least M times, the error is further reduced. If the coefficient K'_α is based on M samples, then K'_α is given by Eq. I-5 where K_{α_i} is the value of the coefficient obtained from the i^{th} sample alone.

$$K'_\alpha = \frac{1}{M} \sum_{i=1}^{M} K_{\alpha_i} \qquad \text{I-5}$$

It is not difficult to show that K'_α converges in the mean to K_α provided the K_{α_i} are statistically independent, which is a reasonable assumption. The variance of the sample mean K'_α is easily shown to be[16]

$$\left\langle (K'_\alpha - K_\alpha)^2 \right\rangle = \frac{\overline{\left[y(t + T) - K_\alpha\right]^2 \; S(\alpha)}}{M \; \overline{S(\alpha)}}$$

The term on the right is merely $1/M^{th}$ of the variance of the output when the input is in state α.

Averaging over all α, the result is

$$\left\langle \sum_\alpha (K'_\alpha - K_\alpha)^2 \; \overline{S(\alpha)} \right\rangle = \left\langle E - E_{min} \right\rangle = \frac{1}{M} \sum_\alpha \overline{\left[y(t+T) - K_\alpha\right]^2 \; S(\alpha)}$$

or:

$$\left\langle E - E_{min} \right\rangle = \frac{1}{M} E_{min}$$

That is, the additional MSE due to a finite input record is no more than $(1/M)\ E_{min}$ if each sequence occurs at least M times in the record.

Errors in the Table Lookup Model

To obtain an idea of the errors inherent in models of this type, the case of a linear system with two level inputs will be studied. An expression for the mean square error will be obtained in closed form. The input process is assumed to have random zero crossings which are Poisson distributed.

In general, for any linear fixed parameter system, the mean square error (MSE) can be computed using the convolution integral:

$$y(t) = \int_0^\infty h(\tau)\, x(t-\tau)\; d\tau$$

The model output for the table lookup scheme is the average value of y given the values of the input at the various sampling points:

$$\overline{y}^{x*} = \int_0^\infty h(\tau)\, \overline{x}^{x*}\; (t-\tau)\; d\tau$$

where $\overline{}^{x*}$ denotes the expected value conditional on the input samples x*.

The MSE is then given by

$$E = \int_0^\infty\!\!\int_0^\infty h(\tau)\, h(\sigma)\, \phi_{nn}\,(t-\tau,\ t-\sigma)\; d\tau\;\, d\sigma$$

where

$$\phi_{nn}\,(\tau',\ \sigma') = \phi_{xx}\,(\tau'-\sigma') - \overline{\overline{x(\tau')}^{x*}\quad \overline{x(\sigma')}^{x*}}$$

where the ——— indicate average over all x*.

The first term on the right in the above expression is the autocorrelation function of the input. The second term is the autocorrelation of the average values $\overline{x(\tau')}^{x*}$ given the sampled values of the input x*.

For a random telegraph wave of amplitude ± 1, the correlation function is given by

$$\phi_{xx}\,(\tau', \sigma') = e^{-2a\,|\tau' - \sigma'|}$$

where the time constant, which is the average time between zero crossings, is 1/a.

Now the probability of having k zero crossings in a length of time T is given by the Poisson probability distribution:

$$P\,(k, T) = \frac{(aT)^k}{k!}\; e^{-aT}$$

So that the probability of an even number of zero crossings in time T is

$$P_e(T) = \sum_{k=o}^{\infty} \frac{(aT)^{2k}}{(2k)!}\; e^{-aT} = e^{-aT}\; \cosh aT$$

Whereas the probability of an odd number is

$$P_o\,(T) = e^{-aT}\,\sinh aT$$

Let $\tau' = t - \tau$, $\sigma' = t - \sigma$; and for convenience, drop the primes.

The function $\overline{x^{x^*}}(\tau)$ is easily determined. It must be done for two cases:

(a) $\tau > \sum_{i=1}^{N} T_i$

(for times further in the past than the last sample point)

(b) $\tau < \sum_{i=1}^{N} T_i$

(between two of the input samples)

It is convenient to let $\tau' = \tau - \sum_{i=1}^{k} T_i$; k = 0,..., N and drop the prime as before. Let $\sum_{i=1}^{k} T_i = I_k$. For case (a), k = N and

$$\overline{x^{x*}}(\tau) = x\left[(N+1)T\right] \quad e^{-a\tau}\left[P_e(\tau) - P_o(\tau)\right]$$

or

$$\overline{x^{x*}}(\tau) = x\left[(N+1)T\right] e^{-2a\tau}$$

and

$$\overline{x^{x*}}(\tau)\,\overline{x^{x*}}(\sigma) = e^{-2a(\tau+\sigma)}$$

from which:

$$\phi_{nn}(t-I_N-\tau,\ t-I_n-\sigma) = 2e^{-2a\max\{\tau,\sigma\}} \quad \sinh\left[2a\min\{\tau,\sigma\}\right]$$

for

$$\tau > 0, \quad \sigma > 0$$

For case (b) k = 0, 1,..., N-1 and the interval to be considered is the k^{th} tap interval. Now there are several cases depending on the values of the input at the k^{th} and $k + 1^{th}$ tap.

If they are the same value (± 1) then the conditional probability that $x(\tau)$ is the same as the value at the tap points is:

$$\frac{P_e(\tau)\,P_e(T_k-\tau)}{P_e(T_k)}$$

Whereas the probability that it is different is:

$$\frac{P_o(\tau)\,P_o(T_k-\tau)}{P_e(T_k)}$$

This gives

$$\overline{x^*}(\tau) = \pm \frac{\cosh a\tau \ \cosh a(T_k-\tau) - \sinh a\tau \sinh a(T_k-\tau)}{\cosh a T_k}$$

or

$$\bar{x}'(\tau) = \pm \frac{\cosh a(T_k-2\tau)}{\cosh a T_k}$$

similarly, if the values of x are different at the taps.

$$\bar{x}^{\circ}(\tau) = \pm \frac{\sinh a(T_k-2\tau)}{\sinh a T_k}$$

To correlate this over all possible combinations:

$$\overline{\overline{x^*}(\tau)\,\overline{x^*}(\sigma)} = \bar{x}'(\tau)\,\bar{x}'(\sigma)\,P_e(T_k) + \overline{x}^{\circ}(\tau)\,\overline{x}^{\circ}(\sigma)\,P_o(T_k)$$

This comes to:

$$\overline{x^*}(\tau)\,\overline{x^*}(\sigma) = \frac{\cosh 2a(T_k-\tau-\sigma) - e^{-2aT_k}\cosh 2a(\tau-\sigma)}{\sinh 2a T_k}$$

or

$$\phi_{nn}(t-I_k-\tau,\ t-I_k-\sigma) = e^{-2a|\tau-\sigma|} - \frac{\cosh 2a(T_k-\tau-\sigma) - e^{-2aT_k}\cosh 2a(\tau-\sigma)}{\sinh 2a T_k}$$

$$k - 0, 1, \ldots, N-1$$

$\phi_{nn}(\tau, \sigma)$ has now been evaluated for the case where τ and σ occur in the same tap interval. If they occur in different tap intervals, then

$\phi_{nn} = 0$ since the zero crossings are independent in the various intervals, so that the errors are uncorrelated for different intervals.

Thus, the mean square error may be given by the double convolution of $h(\tau)$ with ϕ_{nn} which is evaluated above.

APPENDIX III

MSE for First Order System

If the system to be identified were a low pass filter with $h(t) = e^{-t}$, then evaluation of the MSE is not difficult if the expressions developed in the foregoing appendix are used.

First, the mean square system output is easily found:

$$\phi_{yy}(0) = \int_o^\infty \int_o^\infty h(\tau)\, h(\sigma)\, \phi_{xx}(\tau - \sigma)\, d\tau\, d\sigma$$

or

$$\phi_{yy}(0) = \frac{1}{1 + \alpha}$$

where $\alpha = 2a$ is taken for convenience.

The integral expression for the MSE is evaluated in N + 1 stages, one for each of the tap intervals, and one for the time after the last tap. The entire expression is:

$$E = \int_o^\infty \int_o^\infty h(\tau)\, h(\sigma)\, \phi_{nn}(t-\tau,\ t-\sigma)\, d\tau\, d\sigma$$

where ϕ_{nn} is computed in the previous section. An alternate expression is

$$E = \sum_{k=o}^{N} \int_{I_k}^{I_{k+1}} \int_{I_k}^{I_{k+1}} h(\tau)\, h(\sigma)\, \phi_{nn}(t-\tau,\ t-\sigma)\, d\tau\, d\sigma = \sum_{k=o}^{N} E_k$$

Now, making the same changes in variables as before

$$\tau' = I_k + \tau, \qquad \sigma' = I_k + \sigma$$

and dropping the primes:

$$E_k = \int_0^{T_k} \int_0^{T_k} h(I_k + \tau)\ h(I_k + \sigma)\ \phi_{nn}(t-I_k - \tau,\ t-I_k - \sigma)\ d\tau\ d\sigma$$

and

$$E_N = \int_0^{\infty} \int_0^{\infty} h(I_n + \tau)\ h(I_n + \sigma)\ \phi_{nn}(t-I_N - \tau,\ t-I_n - \sigma)\ d\tau\ d\sigma$$

E_N will be evaluated first.

$$E_N = \int_0^{\infty} \int_0^{\infty} \exp\left[-(2I_N + \tau + \sigma)\right] \left\{ 2 \exp\left[-\alpha \max(\tau, \sigma)\right] \sinh\left[\alpha \min(\tau, \sigma)\right] \right\} d\tau\ d\sigma$$

the result is

$$E_N = \frac{\alpha}{(1 + \alpha)^2}\ e^{-2I_N}$$

where I_N is the length of the line, or the total time between the first and last sample.

The final step is to find the error due to our ignorance of the behavior of the input between taps, which is

$$E = \sum_{k=1}^{n} E_k$$

The expression for E_k is

$$E_k = \int_0^{T_k} \int_0^{T_k} e^{-2I_k}\ e^{-(\tau + \sigma)}\ \phi_2(\tau, \sigma)\ d\tau\ d\sigma$$

where

$$\phi_2(\tau, \sigma) = e^{-\alpha|\tau-\sigma|} - \frac{\cosh \alpha(T_k - \tau - \sigma) - e^{-\alpha T_k} \cosh \alpha(\tau - \sigma)}{\sinh \alpha T_k}$$

The result is:

$$E_k = \frac{e^{-2I_k}}{(1-\alpha)^2 (1 + \alpha)^2} \left\{ (\alpha + \alpha^3)(1-e^{-2T_k}) + 4\alpha^2 e^{-(1+\alpha) T_k} - 2\alpha^2 (1 + e^{-2T_k} - 2e^{-(1+\alpha) T_k}) \coth \alpha T_k \right\}$$

The percent error in mean square (in relation to the mean square output is then

$$E = \frac{\sum_{k = o}^{N} E_k}{\phi_{yy}(0)}$$

The entire expression is of the form

$$E = \sum_{j=1}^{N} e^{-2 \sum_{i=1}^{j-1} T_i} g(T_j, \alpha) + K(\alpha) e^{-2 \sum_{i=1}^{n} T_i}$$

where the T_i are the intervals between samples

It is now desired to find the best that can be done for a given value of α (ratio of system time constant to average switching time of input), and for a given model complexity (number of taps). This involves finding the values of the T_i which minimize E.

If the expression for E is differentiated with respect to one of the T's, say T_k, and the result set equal to zero:

$$\frac{\partial}{\partial T_k} g(T_k) = 2 \left\{ \sum_{j=k+1}^{N} \exp \left[-2 \sum_{i=k}^{j-1} T_i \right] g(T_j) - K \exp \left[-2 \sum_{i=k}^{N} T_i \right] \right.$$

The result is independent of T, through T_{k-1}.

Thus, it is possible to obtain a numerical solution for the T_i's by first finding T_N, then T_{N-1}, etc. down to T_1. The results are given in Table 2-3.

APPENDIX IV

Design, Construction and Test Results of an Adaptive-Predictive Model for Nonlinear Processes with Two-level Inputs.

CONTENTS FOR APPENDIX IV

LIST OF FIGURES (APPENDIX IV)

PART I.

INTRODUCTION

A. Historical Review

For some time, people working with control systems have been confronted with the need for an adaptive, predictive model for non-linear processes which would operate from just the input and output records of any given system. The idea of using a digital orthogonal model to identify and predict non-linear processes with switched two-level inputs was proposed by Roy and DeRusso in 1962.[6] The proposed model observed n points of the process input past by the use of a tapped digital shift register, and used each of the 2^n possible patterns thus observed to define an orthogonal cell. The average value of the process output observed when a given cell was occupied was then taken to be the closest possible approximation to the process output for that input pattern. By limiting the speed at which the input could change, the 2^n cells could be reduced to about 50 cells for n=10, thereby making the memory requirements of the model much more reasonable. The suggested model would adapt to a time-varying process, and various process settling times could be accomodated by varying the speed of the shift register clock.

In 1963, Roy, Miller, and DeRusso suggested an improvement in the memory of the proposed model.[7] By making the memory self-organizing rather than of the fixed-address type, an unlimited number of different input patterns could be observed as long as the total number of patterns did not exceed the memory capacity. This prevented the possibility of observing "unrecognizable" input patterns and also allowed for maximum efficiency in the use of memory locations. The same paper also explored the possibilities and limitations of a predictive model. Such prediction would be accomplished by associating a given process output with a previous process input pattern, thereby making a prediction possible when that input pattern was observed again.

B. Statement of the Problem

The purpose of this project was to construct a practical digital model for non-linear processes with two-level inputs, in accordance with the theories stated by Roy, Miller, and DeRusso.[7] The proposed adaptive model is to have capabilities of identification and prediction, with maximum flexibility of operation, allowing for experimentation with various parameters. The design will be governed in part by the availability of an IBM 650 Memory Drum, and by various economic considerations. Although several modifications are made in the design of the model originally proposed, there were no changes made in the general theory of operation.

The reason for the construction of this model is to provide a useful research tool which can be combined with other devices to form various experimental control systems, as well as being used to check the capabilities of models of this general design.

PART II.

MODEL DESIGN

A. General Design

A 20 bit digital shift register is used to form a delay line for the two-level process input. This register is shifted at a rate determined by the input clock. The input and input past are taken from the shift register at 10 points (via a patch board), and these 10 points are used to form an address ("predict" address). The drum memory is searched for a coincident address, and if one is found the information and weighting number associated with that address are placed in registers. A digital-to-analog conversion is performed on the information, and this analog signal is combined with the process output in a ratio determined by the weighting number. An analog-to-digital converter produces the binary equivalent of the analog signal formed by the weighted combination mentioned above. If the weighting number has not reached a preset limit, it is increased by one, and both the new weighting number and the information from the analog-to-digital converter are placed in memory at the coincident address. Each memory location contains 10 address bits, 6 information bits, 4 weighting number bits, and a check bit which indicates unused memory locations.

If the 10 taps form an address which has not been observed before, the machine analog output and the process output are combined in a 0:1 ratio and the weighting number is set equal to 1. The information from the analog-to-digital converter, the weighting number, and the new address are all placed in memory at a previously unused position.

A second group of 10 points is taken from the shift register (via the patch board) to form the model output address ("identify" address). A second coincidence circuit searches the drum for this address, and if coincidence occurs the information associated with the model output address is placed in a second register. The digital-to-analog conversion of the information in this register is used as the model output. The weighting number associated with this information is not used at this time.

If the "predict" and "identify" addresses are taken from the same 10 taps of the shift register, the model output will identify the process observed. If the "identify" address is taken from a group of ten taps which are of the same configuration as the 10 taps of the "predict" address, but located closer to the start of the shift register, the model output will predict the process observed. In this mode of operation the machine is associating a given process output with an input pattern delayed some time T (T is equal to the number of shift register bits between the two tap patterns times the period of the input clock). For each input pattern the model output will be the average process output observed at a time T seconds after that input pattern was received.

The drum memory of the model is capable of storing up to 240 different addresses (input patterns). The variable input clock may be run at rates of up to 75 pulses per second (6 volt, positive pulses). The binary input to the machine must be between ±5 and ±50 volts. The model quantizes a ±5 volt process output signal into 64 levels (±32 levels).

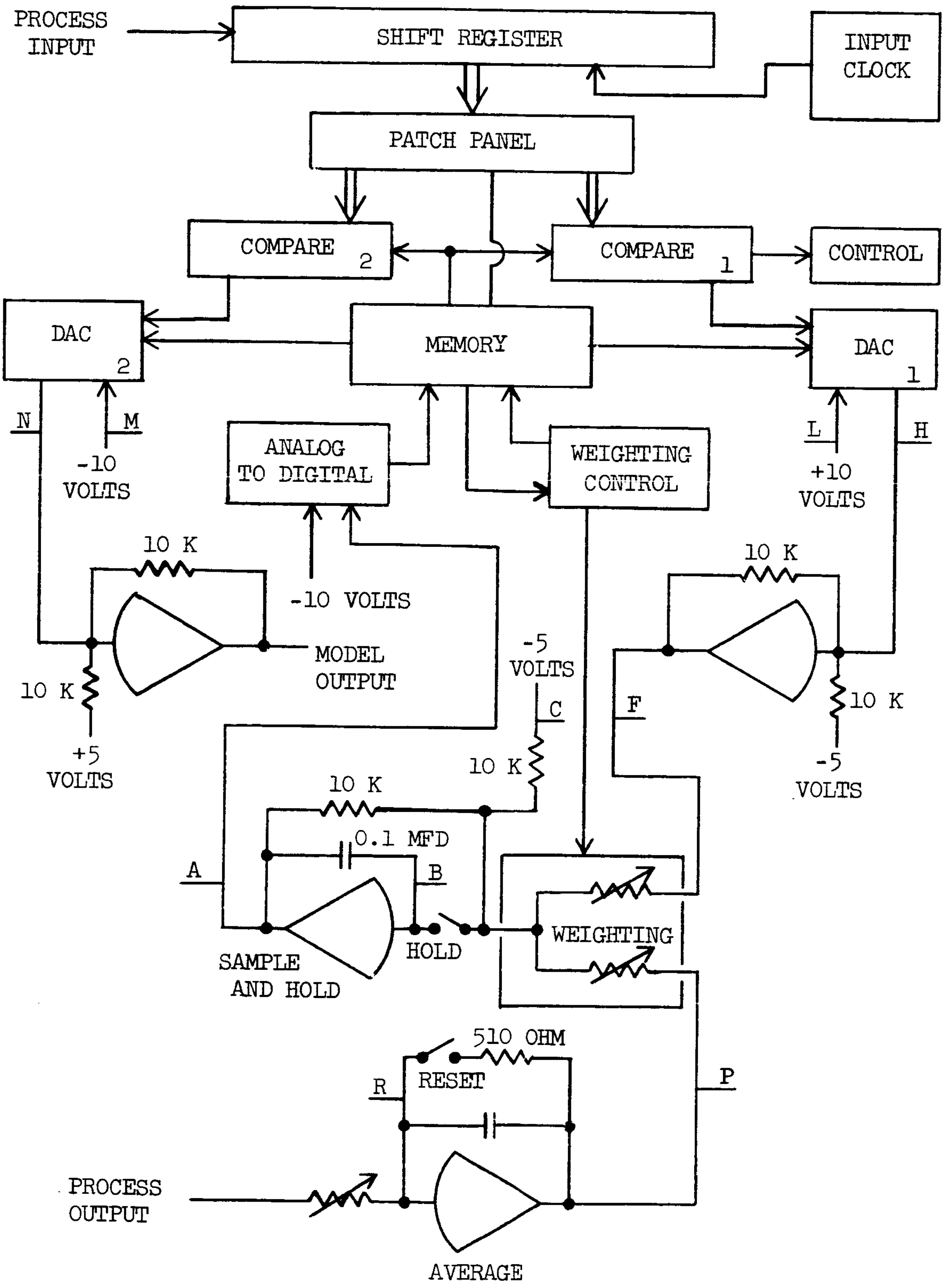


FIGURE 1. BLOCK DIAGRAM

B. Analog Section

The model includes an externally connected analog section, consisting of four operational amplifiers. Although specifically intended for use with an Electronics Associates, Inc. PACE TR-10 analog computer, the model can easily be used with any good quality analog device capable of operating at signals of up to ±10 volts. Reference voltages of ±10 volts and ±5 volts are also required from the analog device.

The block diagram of FIGURE 1 shows the complete circuit of the analog portion of the machine. FIGURE 2 shows the analog connections necessary external to the machine. In both figures the letters refer to pins of the connector located at the upper right of the front panel of the machine. A mating connector is supplied which terminates in color-coded banana plugs. These color-coded plugs are also referenced in FIGURE 2, thus the analog portion of the model may be connected simply by forming the circuits shown.

For optimum operation, it is assumed that a process output of ±5 volts or greater is available. This voltage is attenuated to ±5 volts and used as the input to the process averager shown at the bottom of FIGURE 1. The process averager is an integrator which is sampled and reset to zero once each cycle. If the time between input clock pulses is T, then the average of the input voltage over one cycle is: $e_{average} = \frac{1}{T}\int_0^T e_{input}\, dt$. The equation of the integrator (assuming zero initial conditions) is: $e_{output} = -\frac{1}{RC}\int_0^T e_{input}\, dt$. We shall account for the inversion in later circuits, therefore to obtain an output equal to the average voltage over a cycle, we must set $\frac{1}{RC}$ equal to $\frac{1}{T}$. Any resistor-capacitor combination which is consistent with stable amplifier operation and which has a value RC=T will be acceptable.

The two digital-to-analog converter blocks marked DAC in FIGURE 1 contain digitally controlled relay and resistor networks which serve to connect an input reference voltage to the input of an amplifier through a variety of resistances. These networks are arranged such that when they are connected to the input of an amplifier with a 10,000 ohm feedback resistor, the amplifier output will vary from 0 to 10 volts in 64 steps corresponding to the 64 possible binary numbers in the digital-to-analog converter register. Note that a 5 volt bias signal is added to each DAC amplifier input to change the 0 to 10 volt output to a ±5 volt output. This will cause the binary number 32 in the DAC register to be equivalent to an analog output of zero volts.

The two digital-to-analog converters have opposite polarities. The DAC No. 2 unit which produces the model output is connected for a positive or "correct-polarity" output. The DAC No. 1 unit produces an "inverted-polarity" output. Since the process output has also been inverted (by the process averager), these two signals may now be added directly. The sample-and-hold amplifier will invert their sum, causing a "correct-polarity" signal to enter the analog-to-digital converter.

The weighting unit is another resistor and relay network controlled by digital logic. This network is designed such that when it is connected to an amplifier with 10,000 ohms feedback resistance, the output of the amplifier will be $\frac{e + Xe'}{1 + X}$ where e is the signal from the process averager, e' is the signal from the DAC No. 1 amplifier, and X has integer values from 0 to 15 depending on the binary number in the weighting register. Thus the circuit combines the two inputs in ratios from 0:1 to 15:1.

The amplifier which is used to perform the weighted addition also performs a sample-and-hold function. The 0.1 microfarad feedback capacitor will cause the circuit to remain at a given voltage when the "hold" relay is opened. This is necessary in order to present a non-varying voltage to the analog-to-digital converter circuit during conversion. The capacitor has the added effect of slightly smoothing the input signal (the system has a $\frac{10}{S + 10}$ transfer function), but at the frequencies to be encountered at this point, the effect is negligible. Note that a 5 volt bias is added to the input of this amplifier to convert the ±5 volt inputs to the 0 to 10 volt output which is presented to the analog-to-digital converter. The analog-to-digital converter uses the 0 to 10 volt analog input to produce a 6 bit binary output. Again it should be noted that because of the bias employed, the binary output number 32 actually represents a zero volt signal.

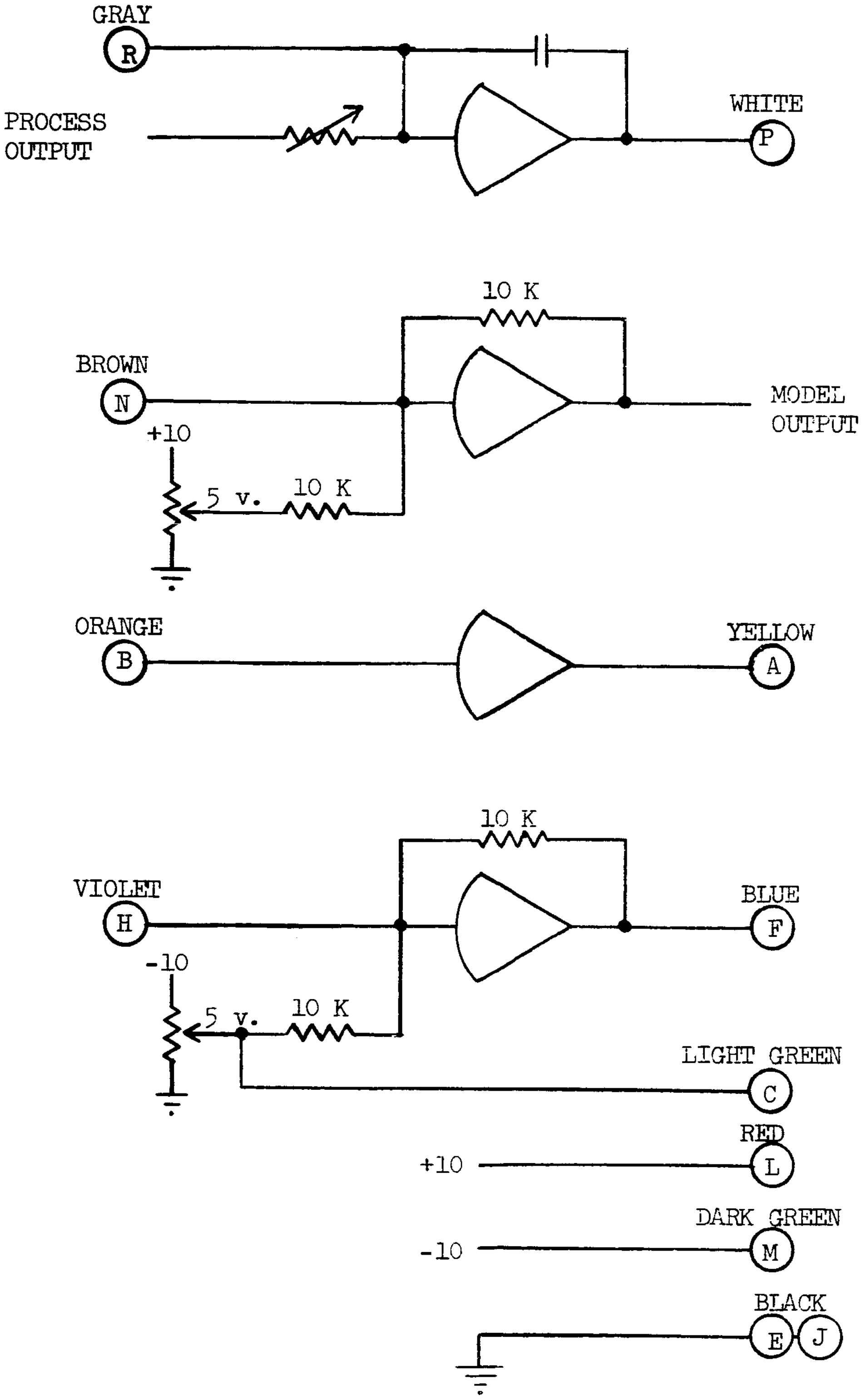


FIGURE 2. ANALOG COMPUTER CONNECTIONS

C. Memory

The memory element of this computer is a modified IBM 650 magnetic drum memory. The modification has been such that instead of the one head per track formerly used, there are 42 tracks available with three heads per track. The drum turns in a clockwise direction as viewed from the rear of the machine, and the three heads on the modified tracks are used for read, bias, and write (from left to right as viewed from the rear of the machine).

The method of recording used is non-return-to-zero.[4, 8] The bias head continually records a "zero" on a given track, and the write head then records a "one" over the "zero" if it is on, or allows the zero to remain if it is off. The read head detects changes between the "one" and "zero" states as positive and negative pulses, which are used to set a flip-flop to either the "one" or "zero" state, corresponding to the information recorded on the drum. The memory is of the circulating type in that the information must be read and re-written each time the drum makes one revolution.

Only 22 of the tracks on the drum are used in the machine (the drum originally contained over 200 tracks, the modification reduced this number to around 140, with 42 of these being accessible from three different heads). Twenty-one of the 42 three-head tracks are used for information storage, and a timing track is used to generate clock pulses at a 125 Kilocycle rate (this pulse rate is divided in the timing logic, so the machine clock frequency is 62.5 Kilocycles). There is a spare timing track (labeled "DP") available in case of damage to the original.

It takes approximately 3.85 milliseconds for information to pass from a position under the write head to a position under the read head. Both the clock frequency and the time necessary for the drum to complete one cycle (from write head to read head) depend on the drive motor speed and will vary somewhat, but one cycle will always contain 240 clock pulses--one clock pulse for each storage location.

The nominal speed of the drum is 12,500 rpm. The main precautions which must be taken in the operation of the drum are to never run the machine when it is cold (below 68 degrees F.), and to oil the drum bearings regularly. The machine contains a timer which gives an alarm every 24 hours of operating time. At this time the bearings should be oiled for 10 seconds by turning the mist-oil motor on and depressing the "oil" button on the power control chassis to the left of the drum for 10 seconds. Over-oiling or under-oiling the drum can both create serious problems and permanent damage. It should be noted that depressing the "drum oil" switch on the front panel will turn off the audible alarm but will <u>not</u> lubricate the drum.

More complete information on the drum and its maintenance is available in the IBM 650 Customer Engineering Maintenance Manual. [2]

PART III.

CIRCUITS

A. Power Control Circuit

The machine requires an input of either two-phase 230 volts or two lines of a three-phase 208 volt system. A 20 ampere, 208-220 volt line-to-line and 115 volt line-to-neutral supply is necessary. To date, all tests were made and all data was taken using a 208 volt input. A ground line is also used to minimize the danger of shock from the machine.

The power control circuit of FIGURE 3 is used to turn the machine on and off, allow the drum to reach running speed before turning on the circuitry, and to indicate the status of the machine power at all times.

The switch S-1 is located on the front of the cabinet, and controls all power to the machine and the convenience outlets located at the top left side of the rear of the cabinet. Closing S-1 will supply power through a 6 volt transformer to the indicator in the "stop" switch on the panel. Pushing either of the start switches P-3 (on the front panel) or P-4 (on the power control chassis) will cause relay K-2 to close and latch, supplying power to the timer T4, to the timer T3, and to the drum motor (through 10 ampere starting fuses). After about 1.5 minutes (time for the drum to attain running speed), the contacts of timer T4 close, causing K-1 to close and latch. This releases K-2, supplies running power to the drum (through a 2.5 ampere fuse), and supplies power to the timer T3, the mist oiler, the demagnetizing unit, and the DC power supply (the latter two units receive power via the duplex outlet on the power control chassis). At this time the indicator in the start switch on the front panel will come on. The DC on-DC off switch on the front panel may now be used to turn on the DC power supply. The DC on indicator will only light if all three DC voltages are present. Therefore if neither the DC on nor the DC off section of the switch is lighted at a time after the drum is at running speed (the start indicator is on), this indicates that either the power switch or one of the circuit breakers on the DC power supply is off. These switches are accessible from the rear of the machine (obviously, a defective indicator lamp could also cause this condition).

If at any time either of the stop switches, P-1 (on the front panel) or P-2 (on the power control chassis) are depressed, or if there is any interruption in the input voltage, all relays will open and the machine will immediately return to the "stop" condition.

The switches S-3 and S-4 are not used in normal operation. Switch S-3 is located to the right of the DC power supply, and consists of two sections which must always be opened and closed together. They are used to disconnect the machine from the power supply output. This is useful when it is desired to use the power supply without turning on the machine (AC input to the power supply may be obtained by removing the power supply plug from the duplex socket on the power control chassis and connecting it to a standard 110 volt outlet). The switch S-4 is located beside the bias circuits on the opposite side of the drum from the power control chassis. It is used to disconnect the

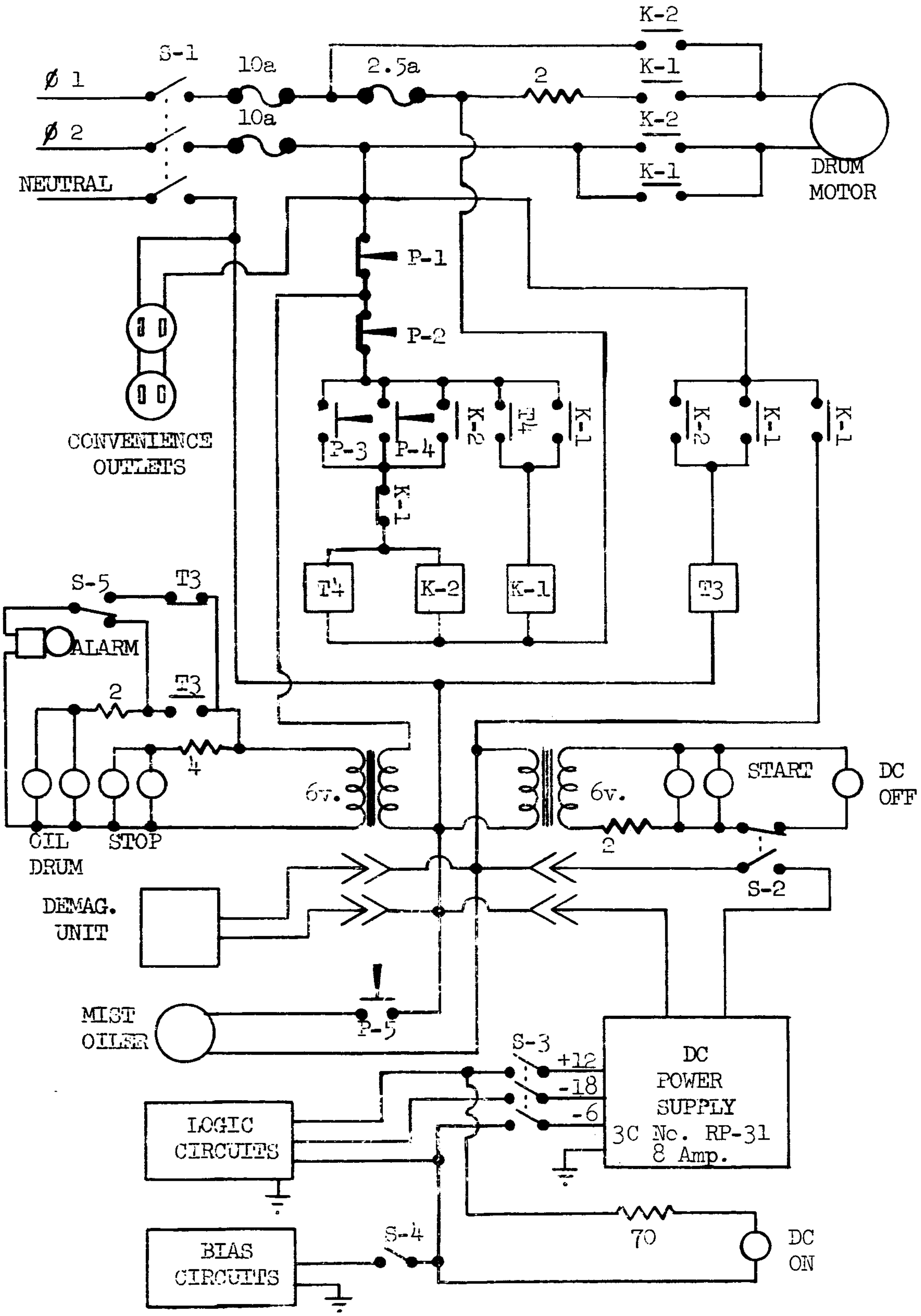


FIGURE 3. POWER CONTROL CIRCUITS

bias circuits when making tests on the logic of the machine without running the drum (with the AC input to the power supply obtained as before). The DC power supply is made by the Computer Control Company, and full information about the unit may be found in the 3C S-Pac Instruction Manual. [1]

The demagnetizing unit is mounted at the right rear of the machine, and can be used to demagnetize a single track or the entire surface of the drum (excluding timing tracks). It is operated by turning the pointer to zero, selecting either "head" or "bar" with the selector switch, and connecting the cable at the side of the unit to a head on the track to be demagnetized (if the "head" position is being used). The power switch on the unit is then held on while the pointer is moved up to full scale and then slowly (over a period of a minute or more) decreased to zero. For more information on the demagnetizing unit, refer to the IBM 650 Customer Engineering Maintenance Manual. [2] There should be no need to use this unit in the normal operation of the machine.

The timer T3 records the drum operating time and operates a signal light and alarm every 24 hours. The switch S-5 (the oil drum switch on the front panel) is used to turn off the alarm, but has no effect on the lubrication of the drum. Drum oiling is accomplished by turning on the switch on the mist oiler motor (mounted on the side of the drum assembly) and then depressing the mist oil pushbutton (P-5) on the power control chassis for 10 seconds. If the alarm has been turned off, it will sound again when the "oil drum" light goes out about five minutes after it came on. Again using the switch S-5, the alarm may be turned off and set to sound at the end of the next 24 hour operating period. It should be noted that under-oiling or over-oiling of the drum bearings can cause serious damage to the drum unit.

B. Logic Circuits

1. General Information

The circuits used all have a "one" or "on" output of zero volts and a "zero" or "off" output of -6 volts. The circuits are all reliable up to 300 Kilocycles or more (excepting the memory circuits), so that at the system clock frequency of 62.5 Kilocycles there are no problems of timing or propagation delays. No fan-in or fan-out problems were encountered. Fan-in of all logical circuits is less than 5, and fan-outs of up to 11 are used with some deterioration of pulse shape, but no loss of reliability at the frequencies in question.

Note that in the logical diagrams an input shown as "1" represents an input which is always on. For the signal voltages used in this machine, these inputs would be connected to ground.

To facilitate testing and repairing of the circuits, each of the 10 large circuit boards are divided into 14 locations. As seen from the front (side on which transistors are mounted), locations 1 through 7 occupy the upper half of the board from left to right, and locations 8 through 14 run from left to right on the lower half of the board. If a location is subdivided, "a" represents the left side and "b" represents the right side. Within or directly under each logic block in the figures is a number which corresponds to the location of that element on the board to which the figure applies. An asterisk (*)

within or directly below a logic block indicates that part or all of that element is located external to the circuit boards. No identification system is used on the plug-in circuits of the memory section, since their small size makes such a scheme unnecessary.

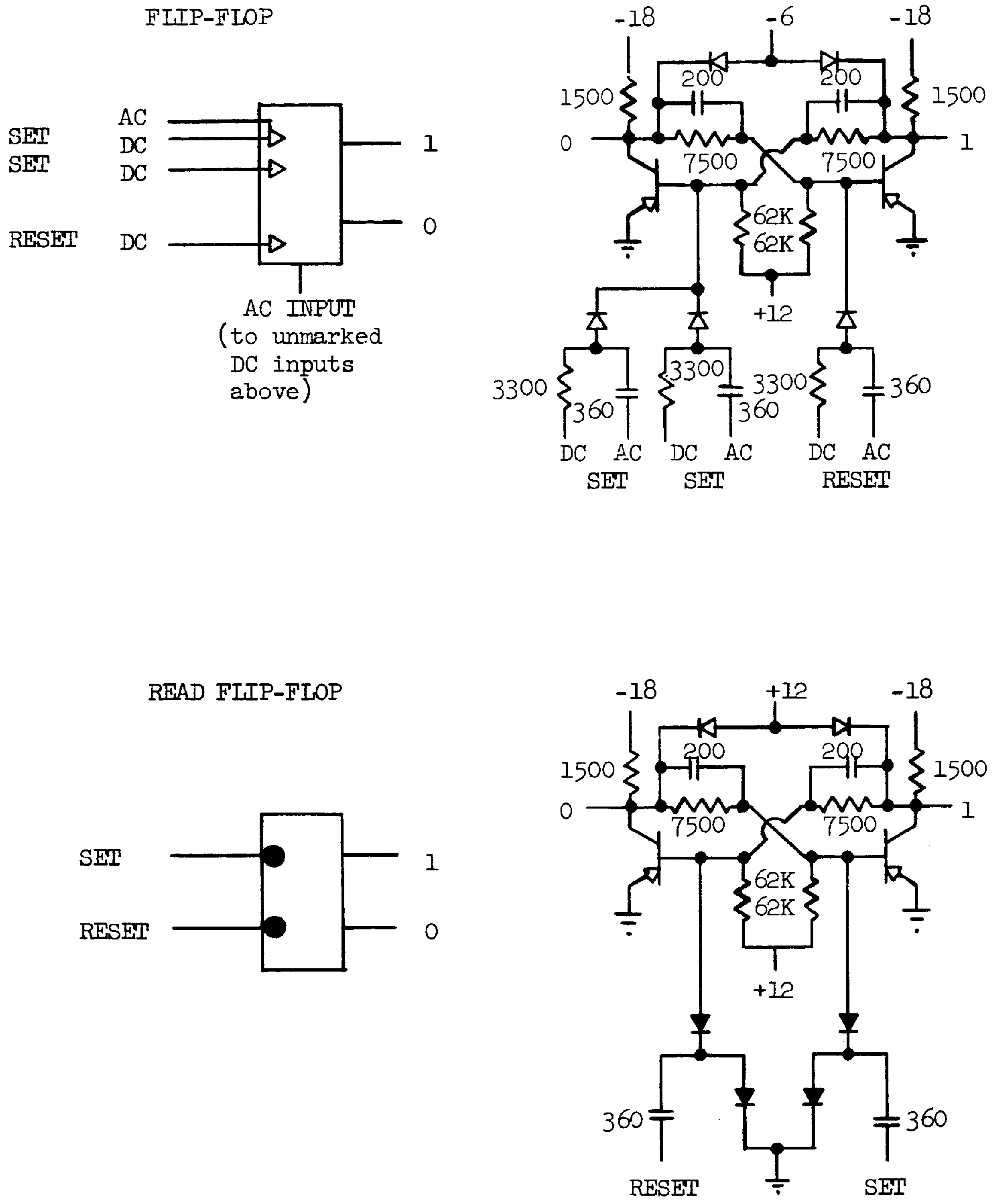
To further facilitate checking, the transistor on the "one" side of all bistable and monostable circuits is marked with a 1 in black ink.

All transistors used in the circuits are RCA type 2N404 germanium PNP transistors. All diodes are germanium type 1N34AS with the exception of those shown as —Ⓓ— which are silicon type 1N482, and those shown as —▶|— which are silicon type 1N673. The 1N482 diodes are used in circuits where their higher current rating is desirable; the 1N673 diodes were used because of their availability.

All resistor values are given in ohms, and all capacitor values are in micro-microfarads unless otherwise specified.

2. Flip-Flop

The standard flip-flop circuit is shown in FIGURE 4. Flip-flops are normally set and reset by gated, positive-going AC inputs to the transistor bases. Up to four of these inputs may be or-ed together at each base, if necessary. Note the two nomenclatures used: one for associating each AC input with a DC gate, the other for associating an AC input with several gates. Flip-flops are in some cases clamped in either the "one" or the "zero" state by connecting the appropriate side of the device to ground through a transistor or switch contact. If more than one flip-flop is thus controlled, diodes are used to isolate each flip-flop from the common line.



FLIP-FLOP

READ FLIP-FLOP

FIGURE 4. BISTABLE CIRCUITS

3. Read Flip-Flop

The read flip-flop of FIGURE 4 is similar to the standard flip-flop, but it is set and reset by negative-going pulses (or by the trailing edge of a positive pulse) at the base of the non-conducting transistor. Diode-capacitor rather than resistor-capacitor inputs are used because of their better reliability with weak signals. These flip-flops are only used in combination with a pair of read circuits.

4. Nand

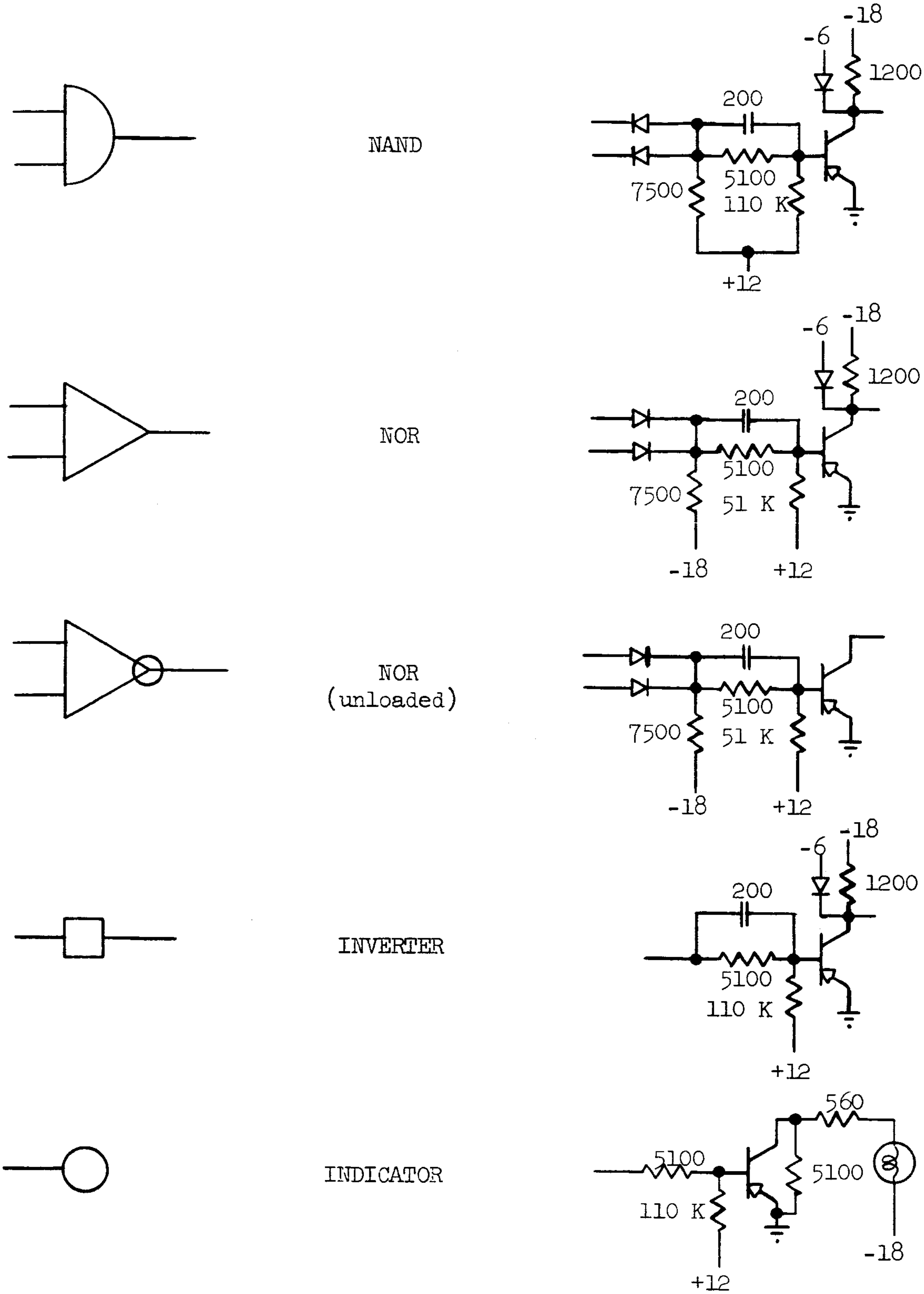
The nand circuit shown in FIGURE 5 consists of a simple diode "and" circuit driving a transistor which provides inversion and level restoration.

5. Nor

The nor circuit shown in FIGURE 5 consists of a simple diode "or" circuit driving a transistor which provides inversion and level restoration. It is sometimes found convenient to obtain an "or" function between the output of a nor circuit and the output of some other circuit (nand, nor, or inverter). This is easily done by connecting the collectors of the transistors of these circuits together. To prevent excessive current in the transistors, only one 1200 ohm resistor load and clamping diode are used. The "unloaded nor" circuit shown in FIGURE 5 is provided for use in this manner.

6. Inverter

The inverter shown in FIGURE 5 is a simple biased transistor which performs the inversion and provides voltage restoration.



Lamp: Dialco 39-10-1435

FIGURE 5. LOGIC CIRCUITS

7. Indicator

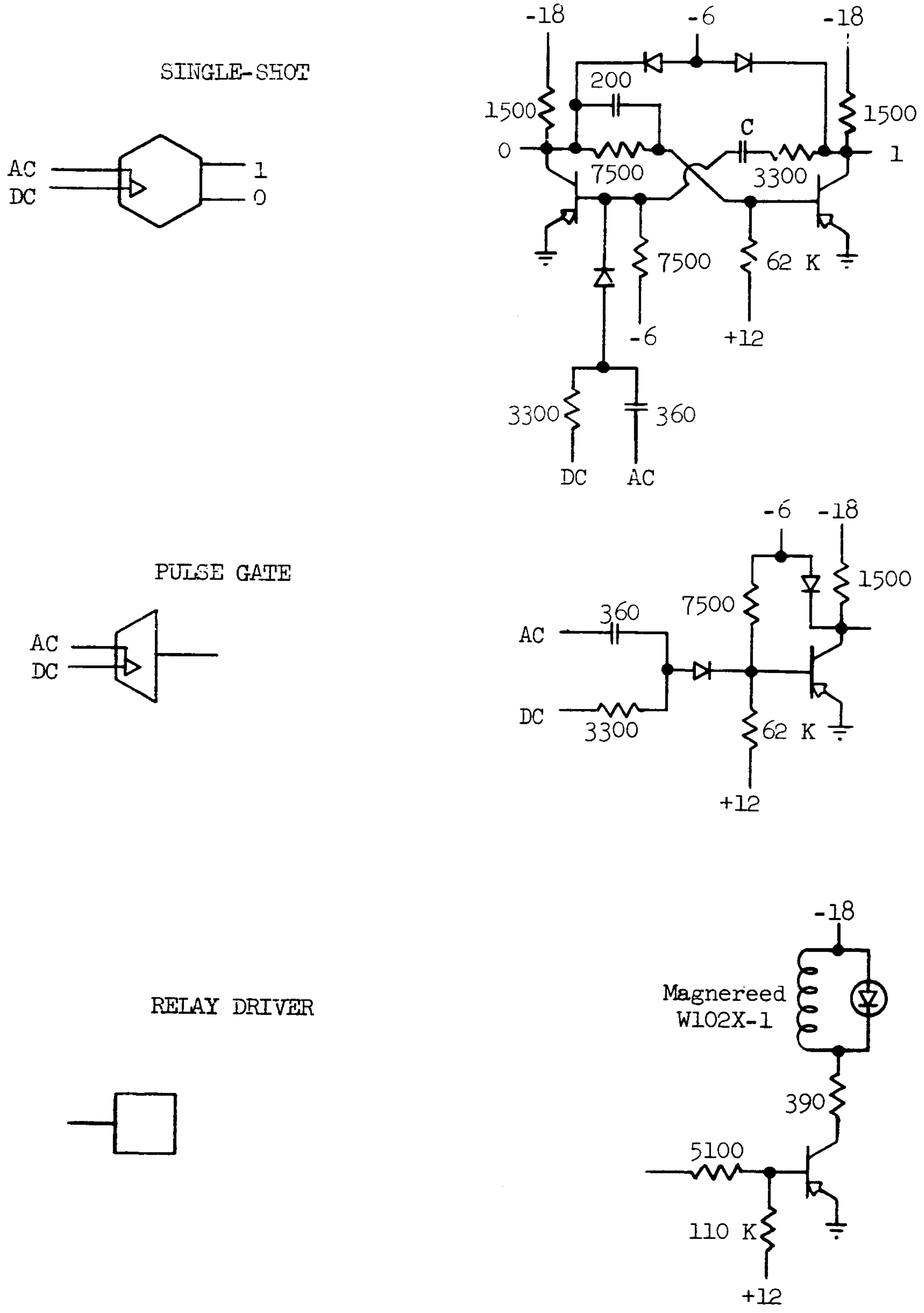
The indicator driver of FIGURE 5 uses a 10 volt, 14 milliampere lamp. A small amount of current is kept flowing in the lamp at all times to eliminate surge current through cold filaments. Note that the indicator driver inverts, thus a "zero" input (-6 volts) causes the lamp to be on.

8. Single-Shot

The single-shot circuit of FIGURE 6 is of common design and is quite similar to the standard flip-flop. When the DC gate is on (at zero volts), a positive-going pulse or level at the AC input will cause the circuit to fire. The timing of the circuit is determined by the value of the capacitor C. For the single-shot used in FIGURE 8, the value of C is 2.0 microfarads and the length of the output pulse is 8.4 milliseconds. For the single-shots used in FIGURE 13 and FIGURE 15, the value of C is 0.2 microfarads and the length of the output pulse is 1 millisecond. Since the "one" output of the single-shot circuit has a poor rise time, an inverter connected to the "zero" output is usually used to obtain a "one" output when it is needed.

9. Pulse Gate

The pulse gate of FIGURE 6 is more or less the first stage of a single shot. When the DC gate input is on (at zero volts), a positive-going pulse applied to the AC input is differentiated by the input capacitor and applied to the base of the transistor. This causes a negative pulse (0 volts to -6 volts, 2 microseconds long) at the output. Because all of the logical circuits operate on positive pulses,



SINGLE-SHOT

PULSE GATE

RELAY DRIVER

FIGURE 6. MONOSTABLE CIRCUITS AND RELAY DRIVER

an inverting element of some kind must follow each pulse gate.

10. Relay Driver

The relay circuit in FIGURE 6 employs a reed-type relay with a SPST normally open contact. Closing time of the relay is less than 1 millisecond, opening time is less than 0.5 milliseconds. The 100 ohm coil draws 32 milliamperes. A type 1N482 diode is shunted across the coil to surpress voltage spikes. Note that the relay will close in the presence of a "zero" signal (-6 volts) at the input to the driver circuit.

11. Input Quantifier

The input quantifier of FIGURE 7 converts any input signals between ±5 volts and ±50 volts to the model logic levels of 0 volts and -6 volts. The input voltage causes a small positive or negative voltage drop across the input diodes. The first transistor is driven between saturation and cut-off by these voltages, and produces levels of 0 and -6 volts at its collector. An inverter is included in the circuit to provide the complement of the input.

12. Read Amplifier

The potentiometer in the read circuit of FIGURE 7 is adjusted such that the transistor is normally off (collector at -6 volts). A negative pulse from the read head will pass through the capacitor and momentarily turn the transistor on, providing a positive pulse which is used to set or reset a read flip-flop (the trailing edge of this pulse is actually used to switch the flip-flop). The resistor in

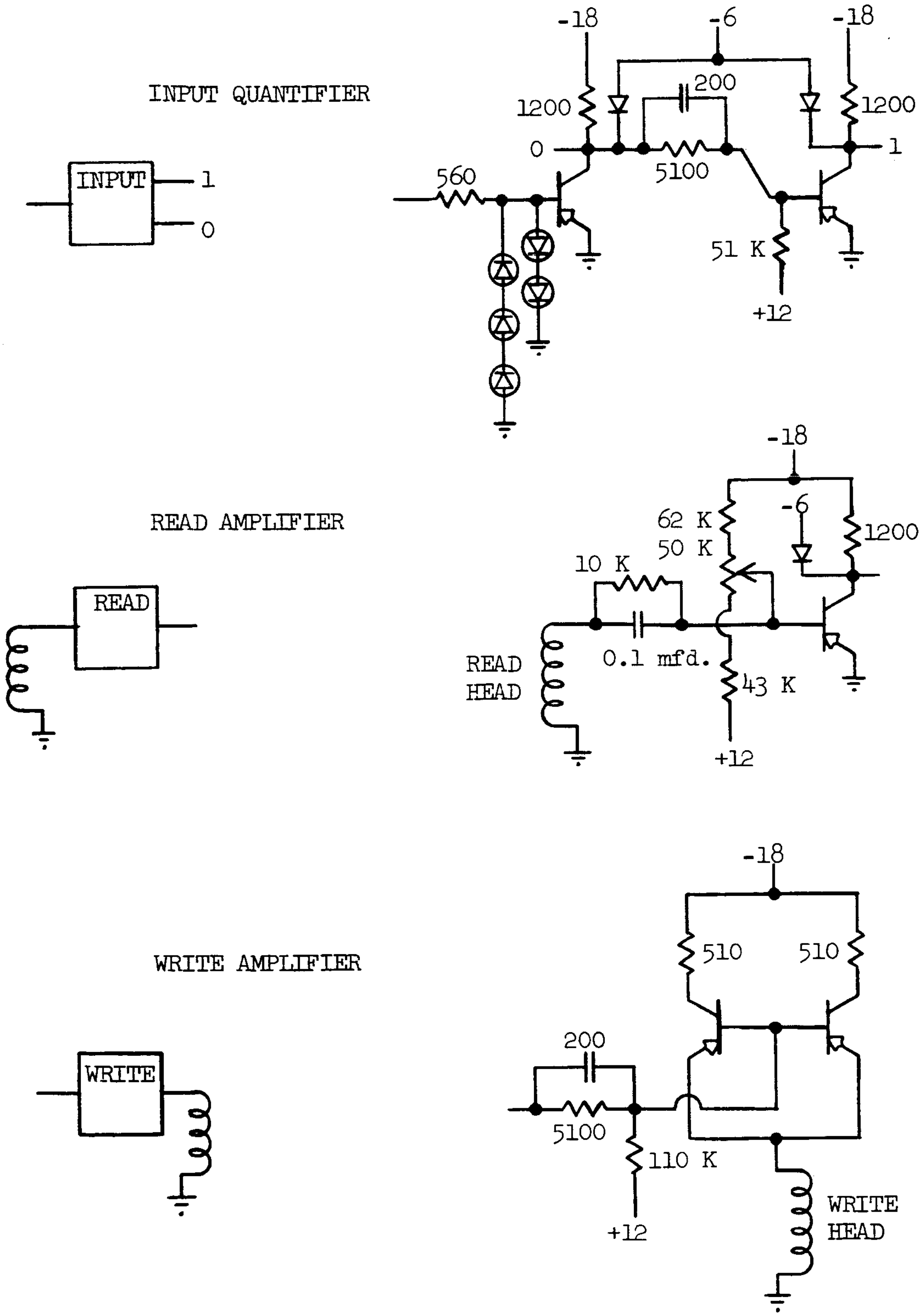


FIGURE 7. SPECIAL CIRCUITS

parallel with the input capacitor is used to surpress noise and spurious signals. Note that this circuit is unaffected by positive pulses at the input. The pulses from the read head should be between 160 and 250 millivolts from peak to peak.

13. Write Amplifier

The write circuit of FIGURE 7 is used to supply current to the write head whenever the write flip-flop associated with it is in the "one" state. Because the circuit inverts, it is connected to the "zero" side of the flip-flop. Two transistors are used in parallel to provide greater current to the head. Associated with each write circuit there is a bias circuit which magnetizes the drum in the opposite direction (see FIGURE 18).

PART IV.

LOGIC

A. Shift Register

Refer to the shift register logic shown in FIGURE 8 and FIGURE 9. The "clock" switch on the front panel allows the operator to select a single clock pulse, no clock pulses, or the external clock pulses (from the input clock). These pulses set the flip-flop S., which enables the single-shot K to fire at the next machine clock pulse (t). S. is also reset at this time. In addition to enabling K, S. also supplies the pulse E which is used to reset the weighting register.

The single-shot K is used to define the period during which the memory is searched for coincidence. It also serves to advance the shift register by one bit. This latter function can be inhibited by placing the "shift register set" switch on the front panel in the "set" position. It should be noted that because this switch clamps the AC inputs of the shift register flip-flops to the "one" state, the act of closing the switch will advance the register by one bit. The "set" switch is used when testing the machine in order to keep a single pattern in the shift register regardless of input changes, while running the remainder of the machine in normal fashion.

The "one" output of each of the 20 bits of the shift register (A_1 through A_{20}) are available at the patch board located on the front panel.

For test purposes, the input to the shift register may be set at "one" or "zero" by using the "input" switch on the front panel.

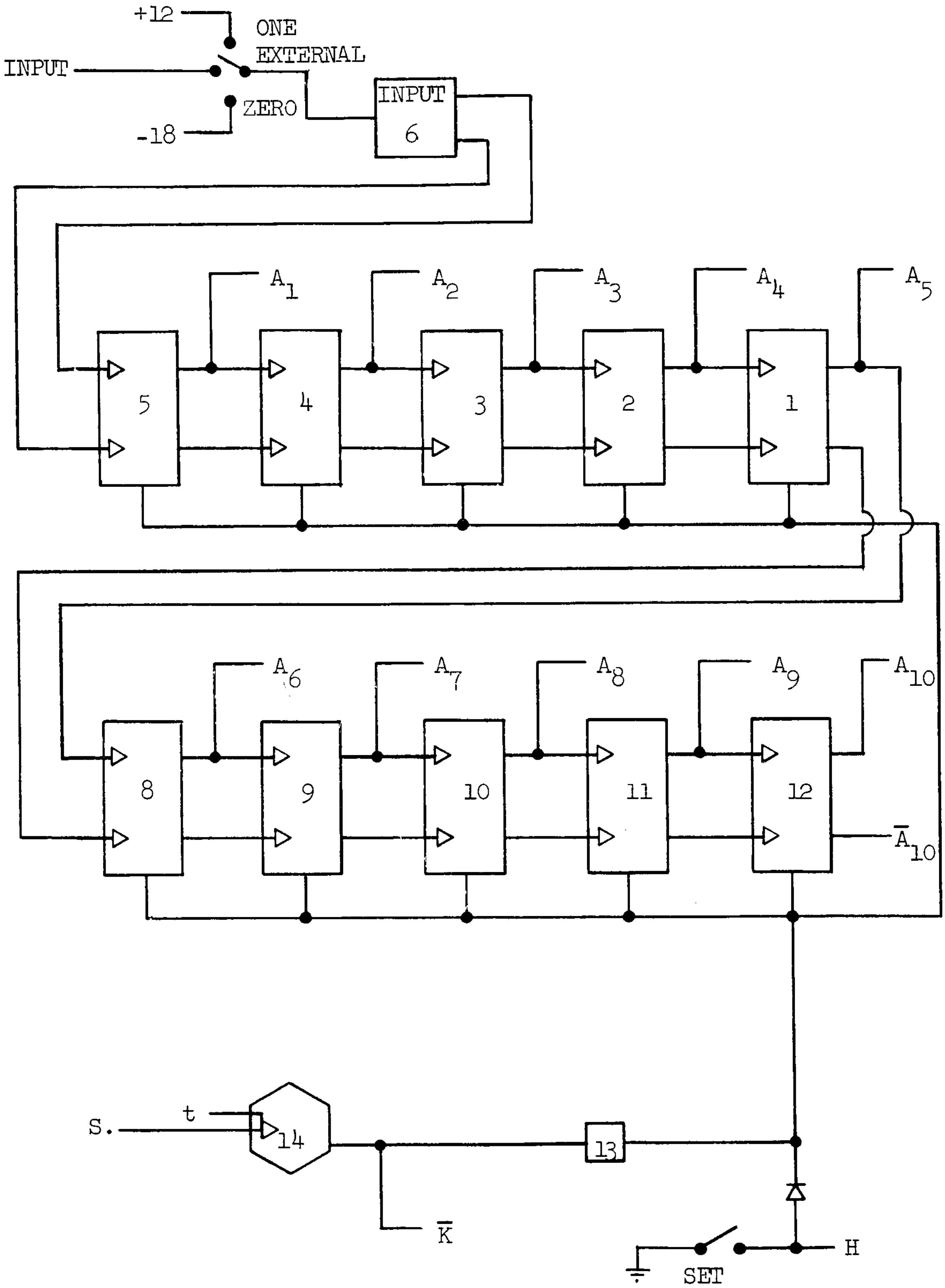


FIGURE 8. SHIFT REGISTER LOGIC, PART 1

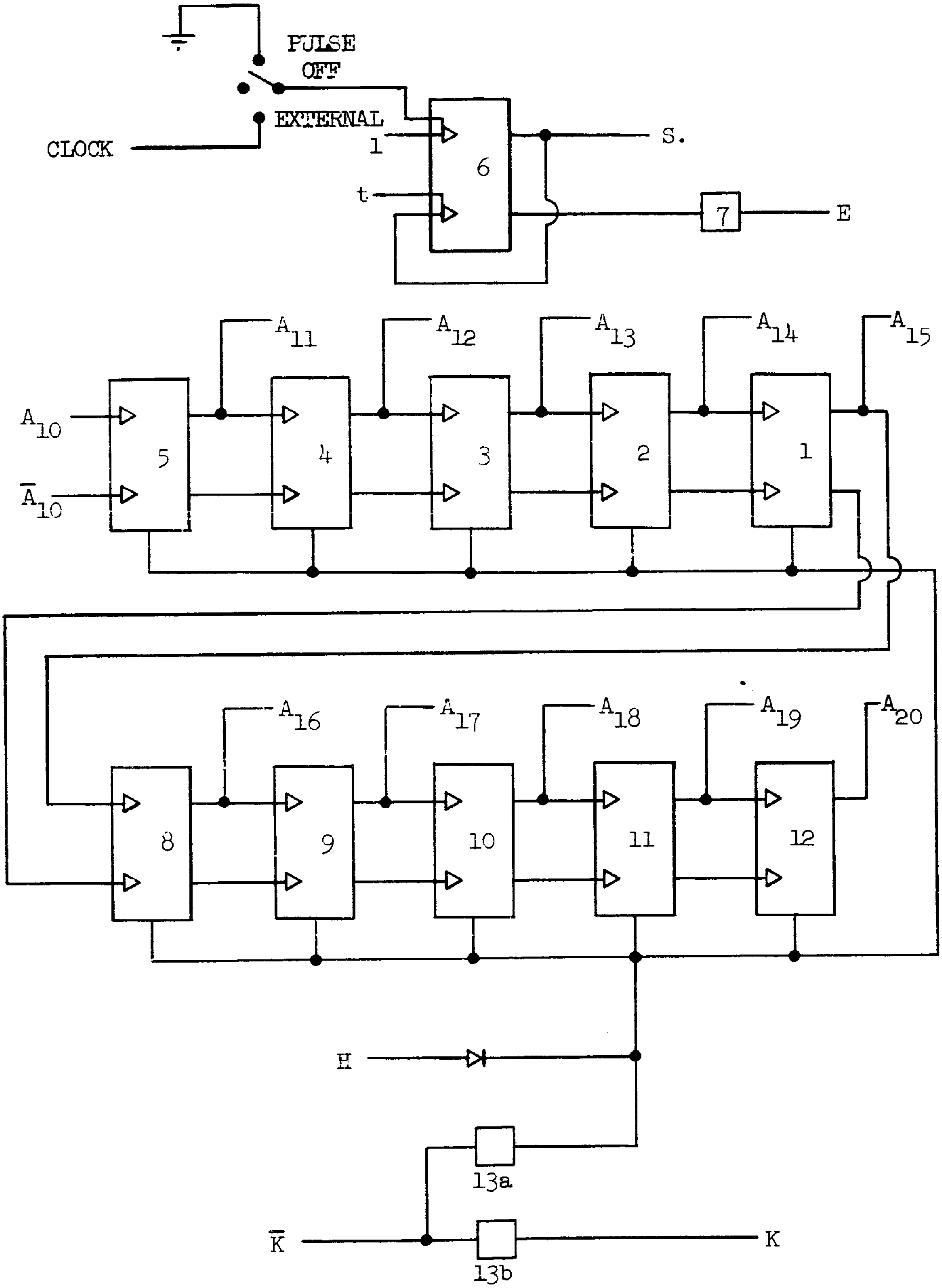


FIGURE 9. SHIFT REGISTER LOGIC, PART 2

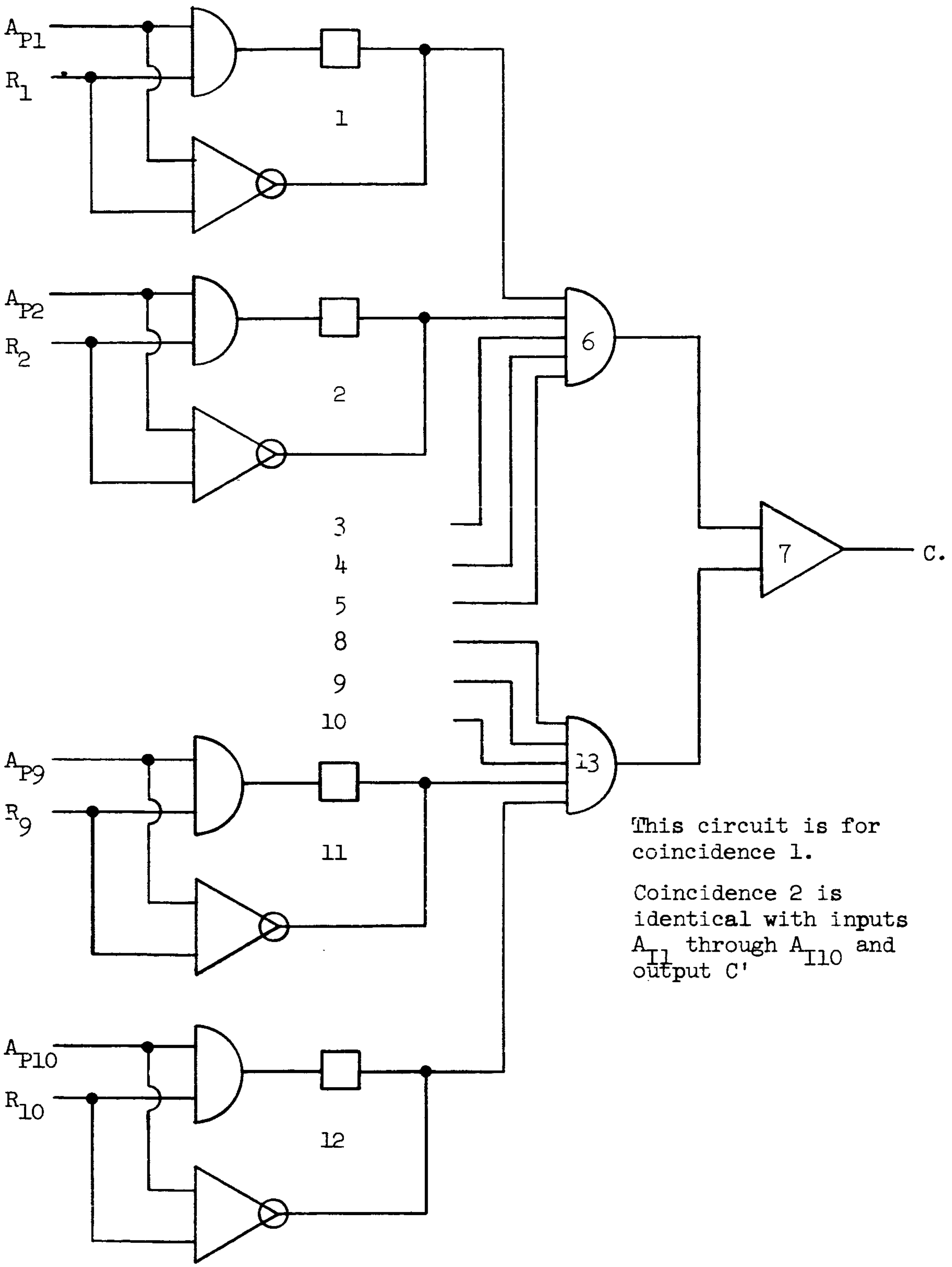
B. Coincidence Logic

Refer to the coincidence logic diagram of FIGURE 10. The patch board is used to select 10 of the 20 shift register points as inputs to the "predict" coincidence circuit (coincidence circuit No. 1). These inputs are labeled A_{P1} through A_{P10}. Similarly, 10 points are selected as inputs to the "identify" coincidence circuit (coincidence circuit No. 2) and are labeled A_{I1} through A_{I10}.

The two coincidence circuits are identical. Each compares the ten input points with the outputs of memory positions 1 through 10 (labeled R_1 through R_{10}) and produces an output if and only if a coincidence occurs between these points. The circuit consists of 10 nand-nor comparison circuits, each of which produces an output if its two inputs are identical. These ten outputs are combined to form a function which is at the "one" level only if each of the ten input pairs are identical. This output function is defined as C. for coincidence circuit No. 1 and as C' for coincidence circuit No. 2.

C. Timing and Control Logic

Refer to the logic diagrams shown in FIGURE 12 and FIGURE 13. The logic in this section of the model is primarily concerned with the flow of information to and from the drum memory. To understand the action of these circuits, it is necessary to consider the way in which information is stored in the memory. Each word in the memory is stored in three parts, which are available at consecutive machine clock times. First there is a check bit (R_{21}) which indicates the presence of a word on the drum as opposed to an unused memory position. Following



This circuit is for coincidence 1.

Coincidence 2 is identical with inputs A_{I1} through A_{I10} and output C'

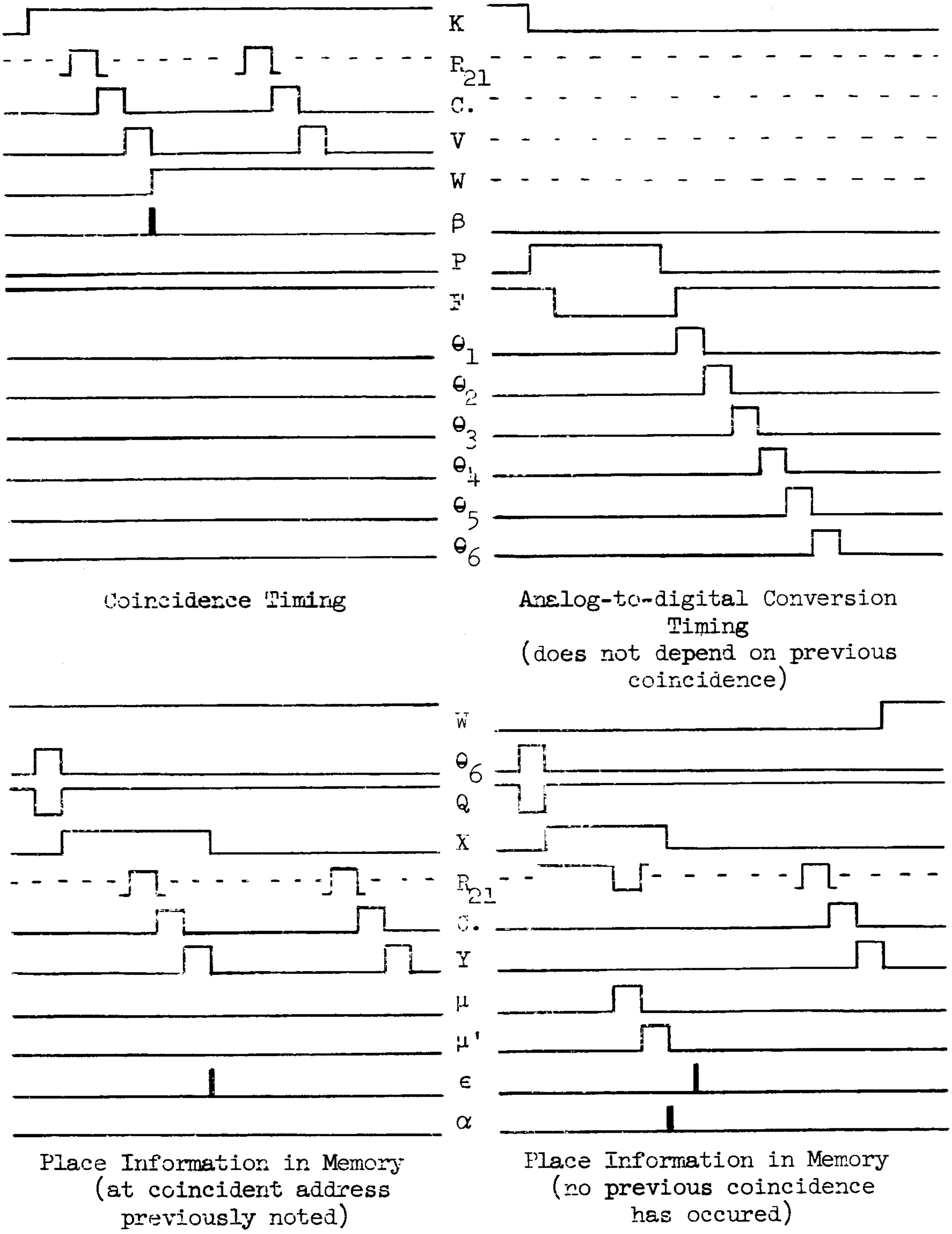
FIGURE 10. COINCIDENCE LOGIC

the check bit by one bit-time is the address (R_1 through R_{10}) which represents a particular shift register pattern (as observed via the patch board). One bit-time later the information (R_{11} through R_{16}) and the weighting (R_{17} through R_{20}) are available.

It is suggested that frequent reference to the timing diagrams of FIGURE 11 will make the sequence of operation of these circuits easier to understand. The timing diagrams are not drawn to scale, and are only intended to indicate the sequence of events, not the duration of those events.

The check bit (R_{21}) is delayed for one bit-time by a flip-flop, and then is and-ed with the output of each coincidence circuit to set flip-flops Y and/or V for one bit-time. Flip-flop Y (controlled by the C' output of coincidence circuit No. 2) controls the flow of information from memory to the digital-to-analog converter circuit No. 2, which produces the model output. Note that this flip-flop is gated such that it only can be set when a coincidence occurs during the time that the signal K is present. This prevents the model output from changing its value as new information from the observed system is assimilated later in the cycle. The flip-flop V (controlled by the C. output of coincidence circuit No. 1) is used to control the flow of information and weighting from memory to their respective registers. Note that V will turn on each time the coincident address appears at R_1 through R_{10}, no matter which portion of the cycle the model is operating in.

The flip-flop W is set by the presence of V, and remains on until a pulse from K indicates the start of a new cycle. W is therefore



Coincidence Timing

Analog-to-digital Conversion Timing (does not depend on previous coincidence)

Place Information in Memory (at coincident address previously noted)

Place Information in Memory (no previous coincidence has occured)

FIGURE 11. TIMING DIAGRAMS

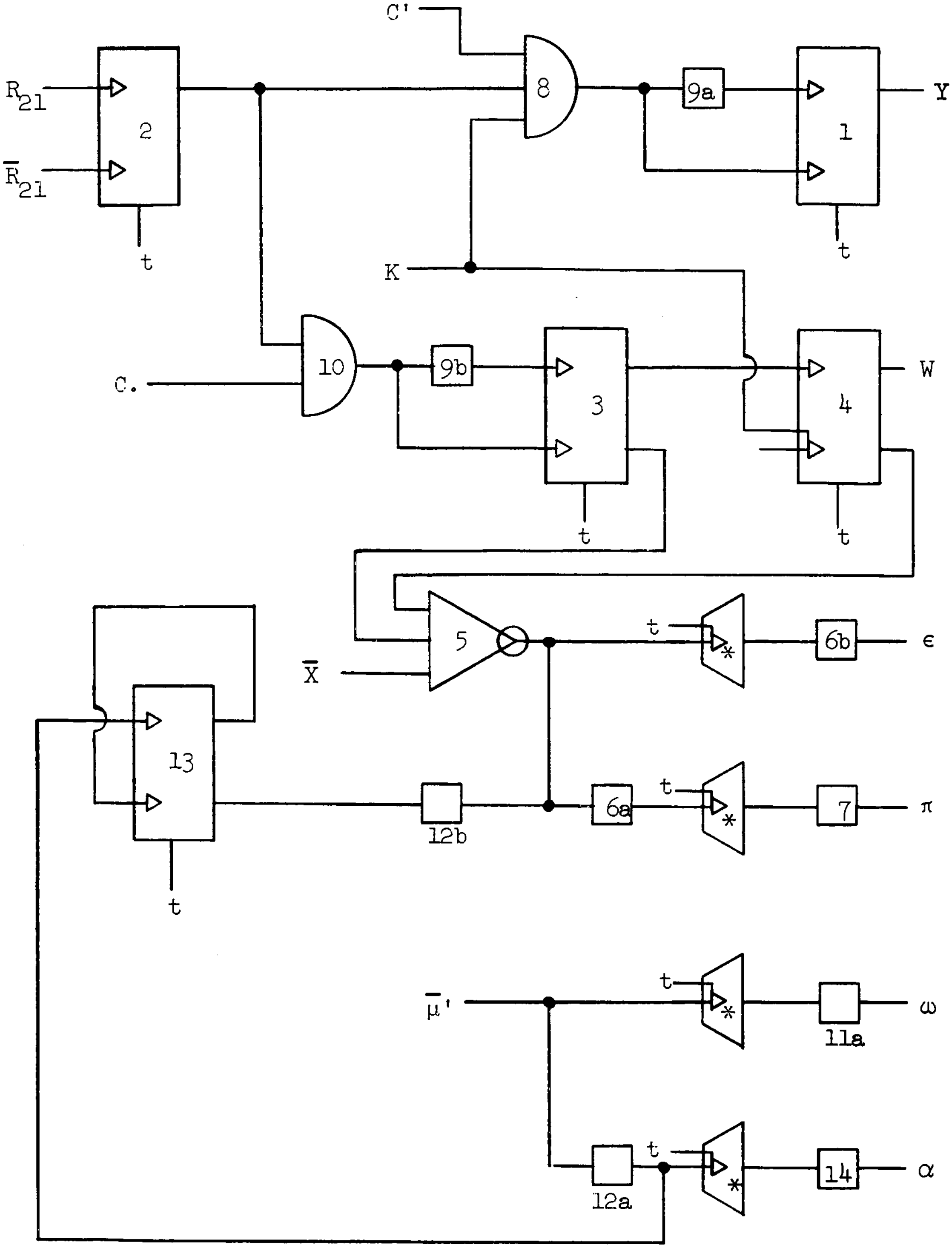
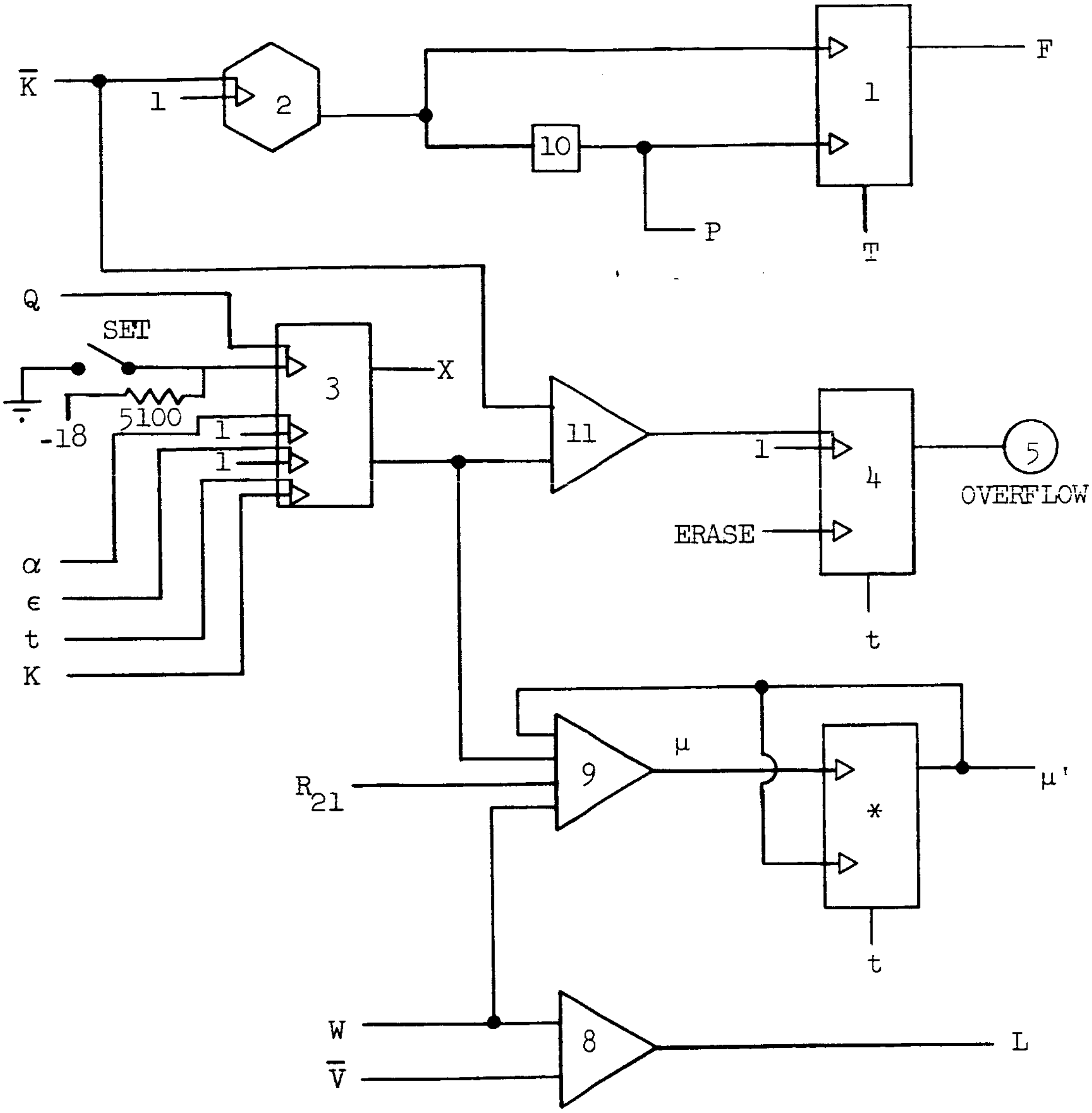


FIGURE 12. TIMING AND CONTROL LOGIC, PART 1



This card also holds ten indicator drivers for the check circuits.

FIGURE 13. TIMING AND CONTROL LOGIC, PART 2

used as an indication that a coincidence has occured during the cycle.

The signal L (equal to the function $V\overline{W}$) is used to control the flow of information and weighting from the proper position in memory to the digital-to-analog converter circuit No. 1 and the weighting register. This is only done at the first coincidence detected in a cycle, due to the fact that $\overline{W}$=0 (W=1) at all times after that.

At the end of the search period determined by the single-shot K, the leading edge of the $\overline{K}$ signal fires single-shot P. P is used to activate the analog hold circuit relay, and provides a delay of 1 millisecond to allow for the relay operating time. The signal P is also used to reset flip-flop F (note that flip-flop F is timed by the T clock pulses used by the analog-to-digital converter, rather than by the usual t machine clock pulses). When the single shot P turns off, $\overline{P}$ causes flip-flop F to turn on. The leading edge of the F signal is used to start the analog-to-digital conversion process.

At the conclusion of the analog-to-digital conversion, the signal Q is generated, the leading edge of which sets the flip-flop X. This action may be inhibited by the use of the "memory set" switch on the front panel. Since X defines the period during which information may be placed in the memory, inhibiting X causes the memory to remain unchanged while allowing the rest of the model to continue operating in a normal fashion.

The flow of information in the memory is controlled by the four gated timing pulses ω, α, ϵ, and π. ω and α control the memory positions 1 through 10 (the address), and π and ϵ control the memory positions 11 through 20 (information and weighting). A pulse ω causes the memory to recirculate information from read to write with no change, while a pulse α causes the information contained on lines A_{P1} through A_{P10} to be inserted in positions 1 through 10 of the memory as a new address. Similarly, a pulse π causes the memory to circulate information from read to write with no change, and a pulse ϵ causes the information in the analog-to-digital converter and the weighting register to be inserted in positions 11 through 16 and 17 through 20 of the memory respectively.

If no previous coincidence has occured during the cycle (a new address is to be placed in the memory), W will be off, and if at some time after X has been set a lack of a check bit (R_{21}) occurs, the signal μ will be formed. μ causes a check bit to be written in memory (signifying that this memory position is now being used), and sets the flip-flop μ'. μ' inhibits μ to prevent a second check bit from being written and also gates a pulse α to the memory while inhibiting a pulse ω. At the next bit-time, the μ' flip-flop resets itself and also sets a flip-flop which gates a pulse ϵ to memory while inhibiting a pulse π. The pulse α causes the flip-flop X to reset, and the logic performs no further functions until a new input pulse is received. Note that it is now possible for a coincidence to occur between the A_{P1} through A_{P10} inputs and the new address in the memory, but since both K and X are in the "zero" state, this will have no effect.

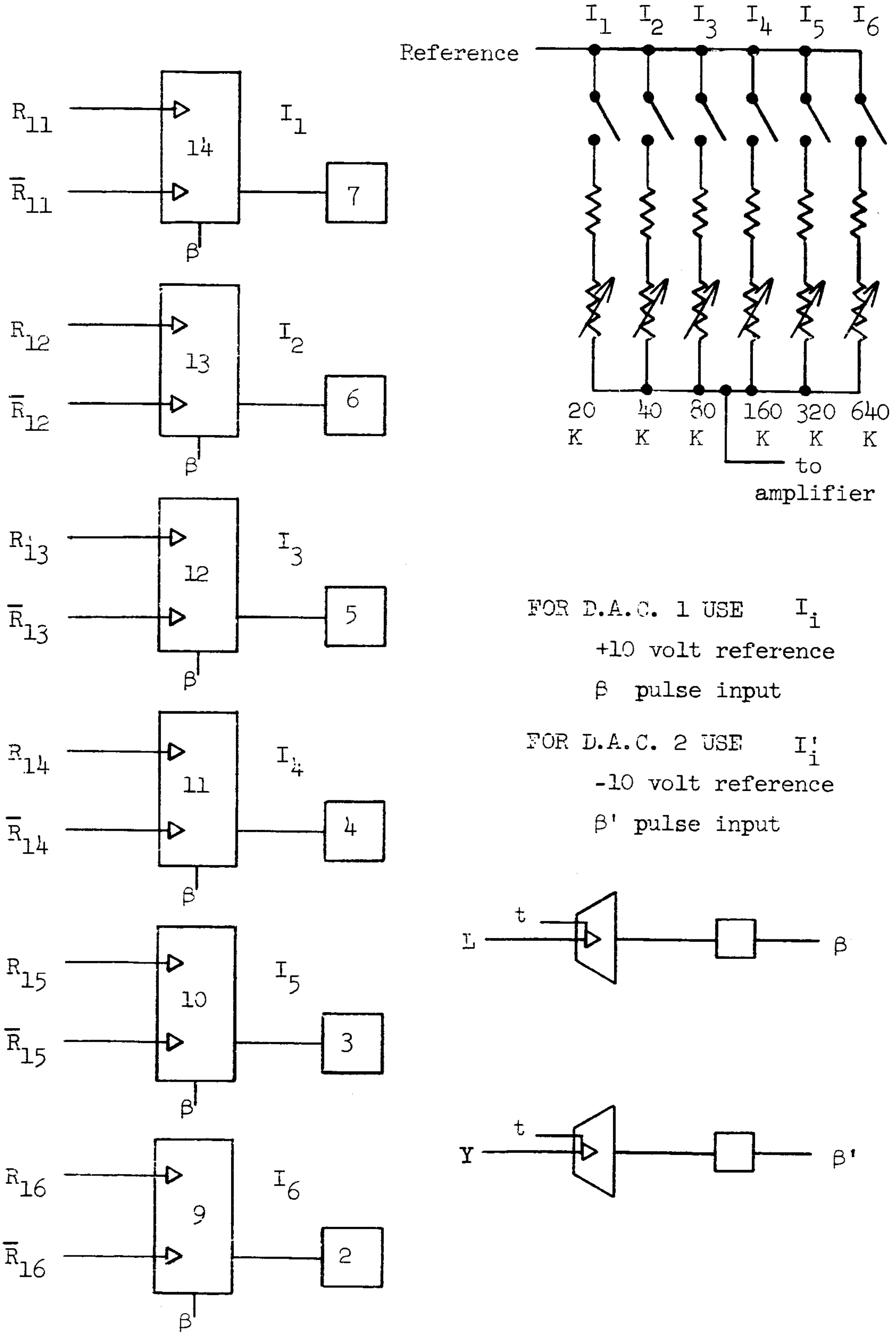
If the flip-flop X is still on when the single-shot K fires again (indicating that no unused memory positions were detected), it is reset by K at the same time that the overflow flip-flop is set by the combination KX. The overflow flip-flop and the indicator which it controls may only be reset by erasing the memory via the "memory erase" switch on the front panel. Notice that the presence of the overflow signal is merely an indication and does not effect the information in memory or the operation of the machine in any way.

If a previous coincidence has occurred during the cycle, then W will be on and if at some time after X has been set a coincidence is detected (indicated by V), then a pulse ϵ will be gated to memory and a pulse π will be inhibited. The pulse ϵ resets the flip-flop X, and the logic performs no further functions until a new input pulse is received (firing K, resetting W, etc.).

D. Digital-to-Analog Converters

Refer to the circuit of FIGURE 14. The digital-to-analog converters each consist of a 6 bit register which controls 6 relays. The relay and resistor circuit is used to vary the input resistance to an operational amplifier with a 10,000 ohm feedback resistor. When the network is fed by a reference voltage, the output of the analog circuit will be a voltage which corresponds to the binary number in the register. The analog connections necessary for these units are shown in FIGURE 1 and FIGURE 2.

The digital-to-analog converter circuit No. 1 uses a +10 volt reference and is used to produce the analog input to the weighting circuit. The register reads information from memory outputs R_{11} through R_{16} when a pulse β is received. This pulse is generated by the signal L which indicates that the information corresponding to the input address



FOR D.A.C. 1 USE I_i

+10 volt reference

β pulse input

FOR D.A.C. 2 USE I_i'

-10 volt reference

β' pulse input

FIGURE 14. DIGITAL-TO-ANALOG CONVERTER

A_{P1} through A_{P10} is at memory outputs R_{11} through R_{16}. The pulse β is also used to cause the weighting register to read the information from memory outputs R_{17} through R_{20}.

The digital-to-analog converter circuit No. 2 is identical, using a -10 volt reference to produce the model output. A β' signal causes the register to read the information from memory outputs R_{11} through R_{16} when a signal Y indicates a coincidence of the information's address in memory and lines A_{I1} through A_{I10}.

Note that these circuits are not bipolar in nature. A 5 volt reference is added to the analog amplifier input of each converter to change the 0 to 10 volt signal to a ±5 volt signal. Therefore, the binary number 32 in the register actually corresponds to a zero volt analog signal.

E. Weighting Logic

Refer to the logic diagrams of FIGURE 15 and FIGURE 16. The weighting register receives information from memory output lines R_{17} through R_{20}. This information is read into the register when a pulse β is received (refer to the description of the digital-to-analog converter circuit No. 1).

The register controls 8 relay drivers which in turn control the inputs to an analog amplifier. The nature of the circuit is such that as the binary number in the register varies from 0 to 15, the output of the DAC No. 1 amplifier and the output of the process averager are combined in ratios varying from 0:1 to 15:1.

The four nor circuits associated with the weighting register are used to add 1 to the number in the register. The four pulses θ_1

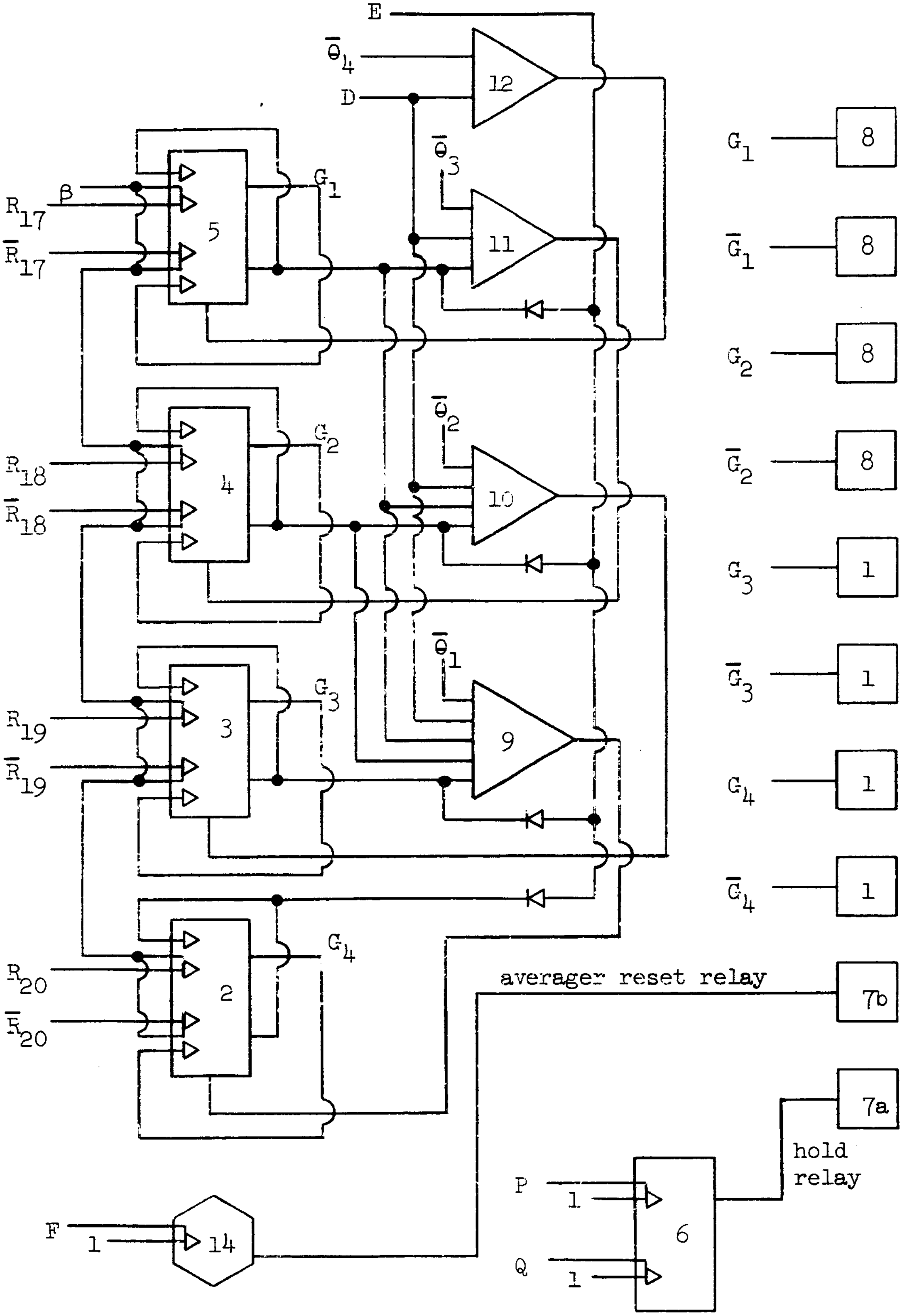


FIGURE 15. WEIGHTING LOGIC

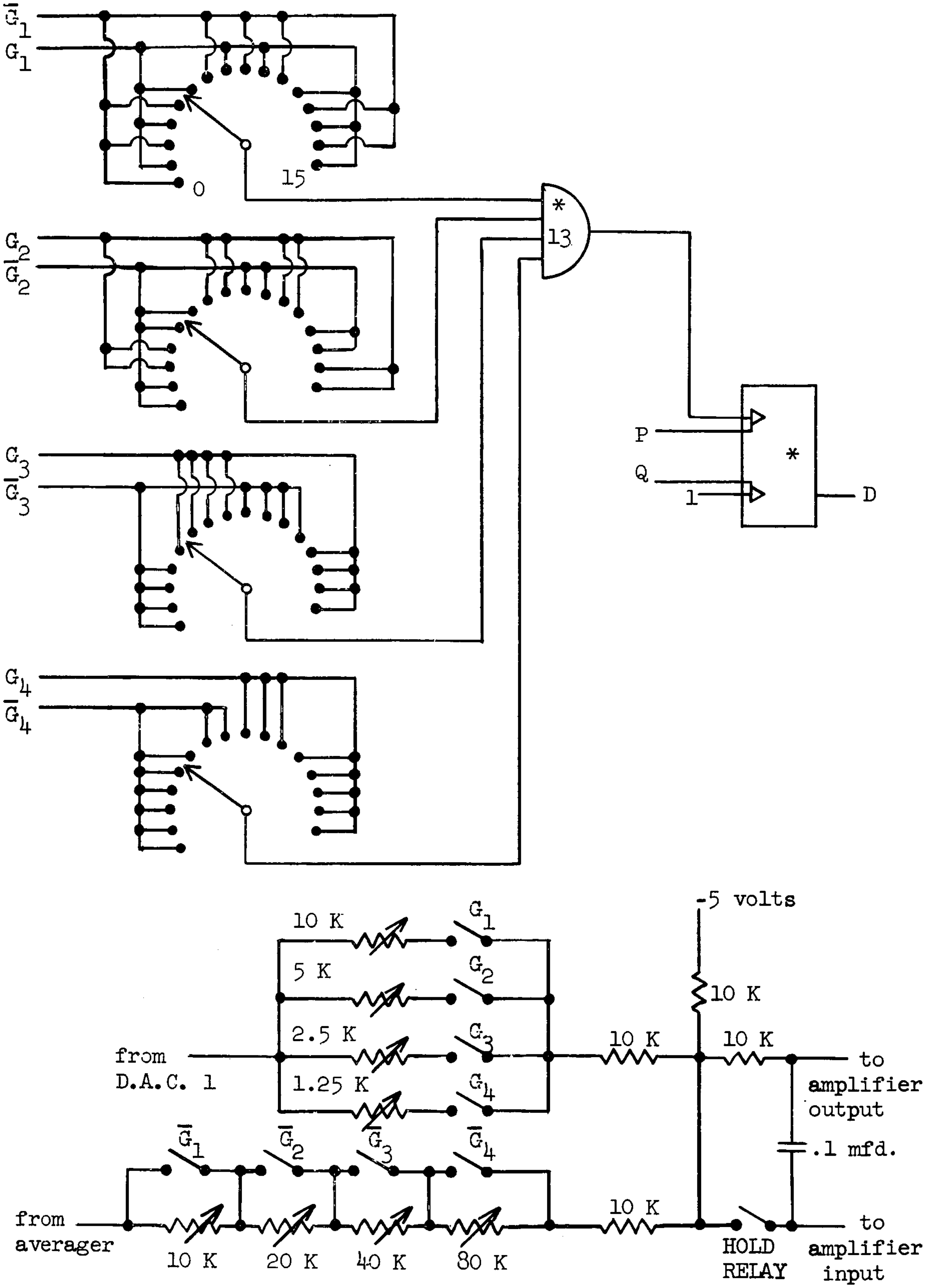


FIGURE 16. WEIGHTING AND LIMIT LOGIC

through Θ_4 (obtained from the analog-to-digital converter logic) arrive in sequence and cause the information in the register to increase by one bit unless the signal D is present. Flip-flop $\overline{D}$ is set by the leading edge of P if the number in the register is not at the value indicated by the weighting limit switch on the front panel. Otherwise it remains off (being reset by Q in the previous cycle), and the D signal prevents the weighting from increasing from the limit value.

At the beginning of each cycle the signal E is used to clamp all four flip-flops of the weighting register to zero. This is done so that if no coincidence occurs the weighting number will have a value of zero and the incoming information will be weighted 1:0.

There are two other circuits in this section: the hold relay circuit and the process averager reset relay circuit. The hold flip-flop is set by the leading edge of P and reset by the signal Q (end of the analog-to-digital conversion). Note that the relay controlled by this flip-flop is normally closed, and opens when the flip-flop is on.

The process averager reset single-shot is used to reset the analog process averager to zero once during each cycle. It is fired by the leading edge of F, and remains on for 1 millisecond to allow the relay to operate and the capacitor to discharge (see the description of the analog circuit).

F. Analog-to-Digital Conversion

The analog-to-digital converter circuit of FIGURE 17 consists of a register, a fast digital-to-analog converter, and a comparator.

The digital-to-analog unit and the comparator were made by the Computer Control Company. The digital to analog unit (LP-31) is a group of 6 electronic switches with a conversion accuracy of 0.1%. The comparator (CD-30) is capable of detecting a 1 millivolt difference between its input and ground. The digital-to-analog converter can be operated at rates up to 200 Kilocycles, the comparator can be operated at rates up to 300 Kilocycles. For further information on these units, consult the 3C S-Pac Instruction Manual.[1]

A ripple down converter technique is used in this section.[3] Flip-flops Θ_1 through Θ_6 form a six bit ring which starts when the signal F is received and stops when Θ_6 returns to the "zero" state (producing the output Q). As each bit of the ring goes on, a "one" is inserted in the corresponding bit of the register. If the number in the register then causes the output of the digital-to-analog converter to be greater than the input analog signal (from the weighting and hold circuits) the output of the comparator gates the reset lines of the register so that the bit in question is reset when the ring advances to the next bit. As the ring advances to the next bit, the next least significant bit of the register is made equal to one, and the comparison process is repeated as before. Thus the digital number is formed by only 6 comparisons.

The comparator is sampled at timing pulse M, while the six-bit ring advances at timing pulse T. The M and T pulses alternate and thus allow the various circuits time to stabilize before proceeding with each step of the conversion. The register is clamped to the "zero" state when single-shot P is on (just prior to the start of the analog-

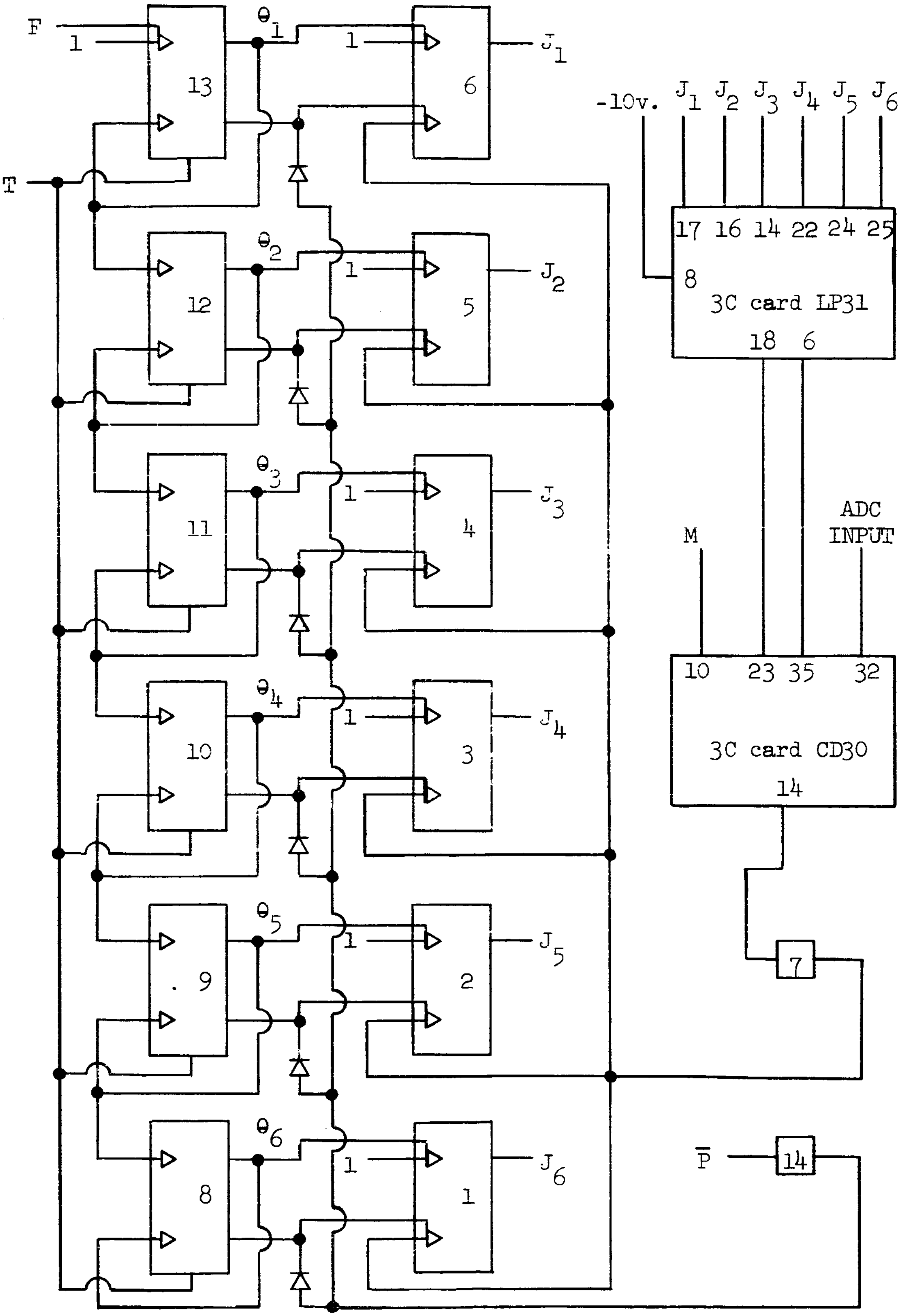


FIGURE 17. ANALOG-TO-DIGITAL CONVERTER

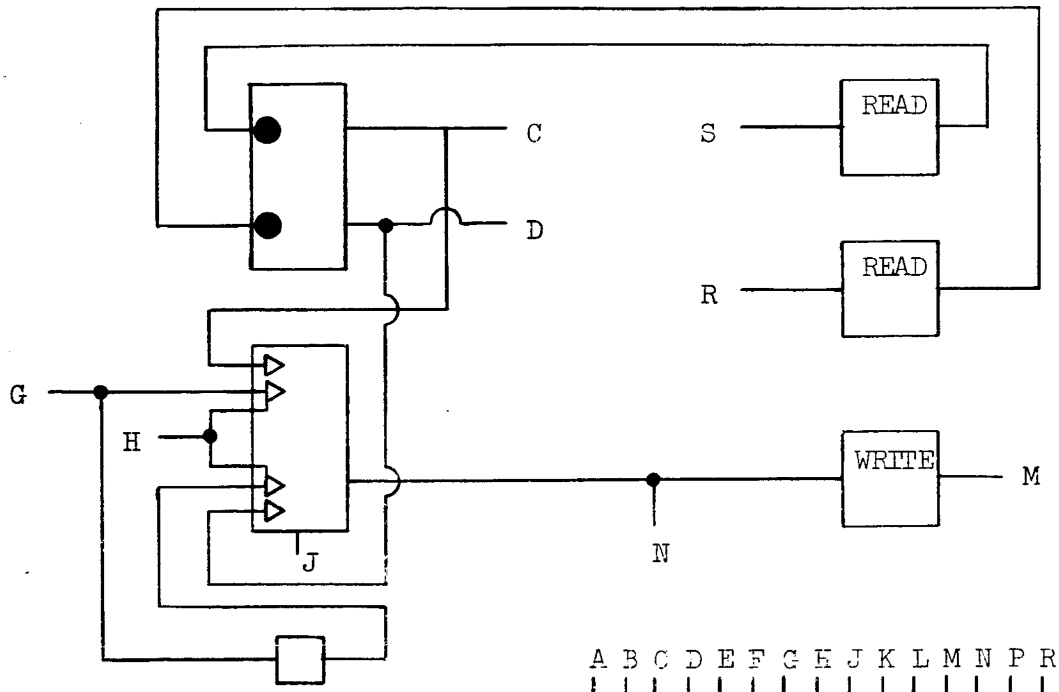
to-digital conversion). This is necessary since the register must start at zero for the conversion to proceed properly.

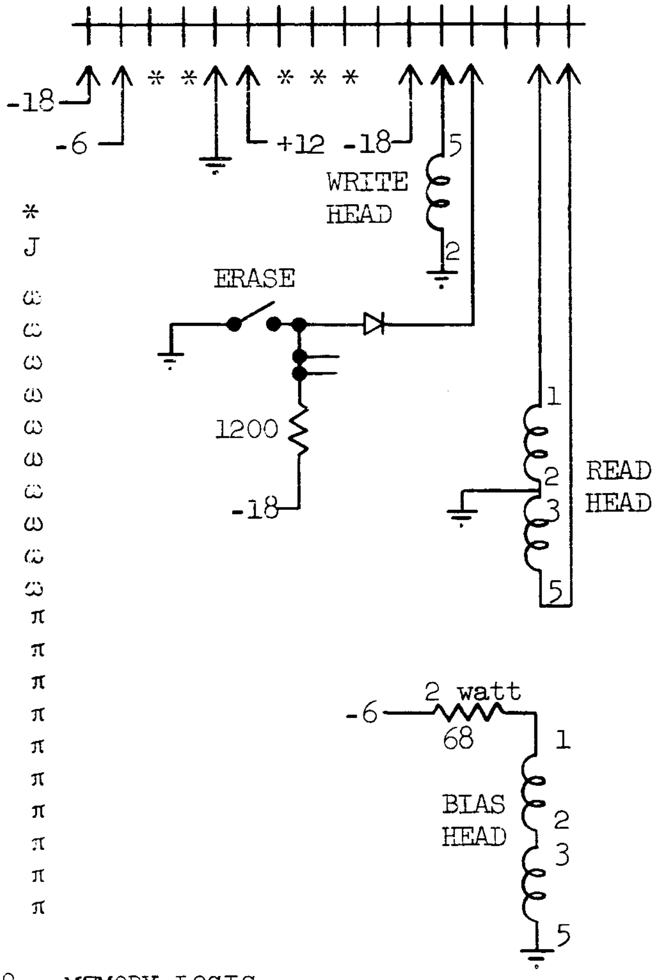
Note that the input to the comparator (from the weighting and hold circuits) is actually a 0 to +10 volt signal, while the output of the 3C digital-to-analog converter is a 0 to -10 volt signal. These are added at the input to the comparator and the sum is compared with ground. Because of biasing used in the analog portion of the model, a ±5 volt signal is converted to the 0 to +10 volt input seen by this device. Thus the binary number 32 in the register actually represents a 0 volt signal, a binary 64 represents a +5 volt signal, and a binary 0 represents a -5 volt signal.

G. Memory Logic

The logic circuits for memory positions 1 through 20 are identical (refer to FIGURE 18). Each circuit performs the function of recirculating information through the memory when timing pulses appear at input J, and inserting new information (present at input G) when a timing pulse appears at input H. Each memory circuit is contained on a plugboard which plugs into one of 20 sockets in an Elco Varipak which is accessible by opening the left door panel of the machine. The sockets in the Varipak are connected to memory positions 1 through 20, from right to left.

The read head associated with each circuit is grounded at its center tap so that flux changes of opposite polarity will appear as negative pulses across opposite sides of the head. Since the read circuits are adjusted so that they are only sensitive to negative pulses,





	*	*	*	*	*
POSITION	C	D	G	H	J
1	R_1	$\bar{R}_1$	A_{P1}	α	ω
2	R_2	$\bar{R}_2$	A_{P2}	α	ω
3	R_3	$\bar{R}_3$	A_{P3}	α	ω
4	R_4	$\bar{R}_4$	A_{P4}	α	ω
5	R_5	$\bar{R}_5$	A_{P5}	α	ω
6	R_6	$\bar{R}_6$	A_{P6}	α	ω
7	R_7	$\bar{R}_7$	A_{P7}	α	ω
8	R_8	$\bar{R}_8$	A_{P8}	α	ω
9	R_9	$\bar{R}_9$	A_{P9}	α	ω
10	R_{10}	$\bar{R}_{10}$	A_{P10}	α	ω
11	R_{11}	$\bar{R}_{11}$	J_1	ε	π
12	R_{12}	$\bar{R}_{12}$	J_2	ε	π
13	R_{13}	$\bar{R}_{13}$	J_3	ε	π
14	R_{14}	$\bar{R}_{14}$	J_4	ε	π
15	R_{15}	$\bar{R}_{15}$	J_5	ε	π
16	R_{16}	$\bar{R}_{16}$	J_6	ε	π
17	R_{17}	$\bar{R}_{17}$	G_1	ε	π
18	R_{18}	$\bar{R}_{18}$	G_2	ε	π
19	R_{19}	$\bar{R}_{19}$	G_3	ε	π
20	R_{20}	$\bar{R}_{20}$	G_4	ε	π

FIGURE 18. MEMORY LOGIC

different directions of flux change passing beneath the head will cause different polarity pulses, which will in turn activate different read circuits.

The read circuits are adjusted such that their outputs are normally at -6 volts ("zero" level), and a negative pulse from the read head causes the corresponding read circuit to produce a short, positive-going, flat-topped pulse. The trailing edge of this pulse causes the read flip-flop to set or reset. Since non-return-to-zero recording is used,[4, 8] the read flip-flop will only change state when two different bits follow one another on the drum. The read flip-flop will not customarily change state at every clock pulse, although it could if the memory track it represented contained a 101010 series.

The input N is used to clamp the write flip-flop in the "zero" state when the erase switch on the front panel is closed. A diode is used between each memory circuit and the erase switch to provide isolation.

The outputs of the read flip-flop appear on pins C and D of the memory card. A complete list of inputs and outputs for circuits 1 through 20 is provided in FIGURE 18. It should be noted that the letters shown in the diagram refer only to pins on the memory circuit card, and not to system signals. Although some pins on the card are shown as being blank, these pins may have been used on the sockets as terminal points.

Associated with every write circuit (positions 1 through 21) is a bias circuit. These circuits are located on the floor of the machine, below the Varipak, and are used to provide a current which constantly writes a "zero" on each memory track. Thus when the write circuit is off, a "zero" appears on the memory track, and when it is on a "one" is written over the "zero" previously recorded.

The numbers on the read, write, and bias heads refer to the connections on the plugs associated with each head. These plugs and their mating sockets are on the memory drum assembly itself, and are only accessible when the cover is removed from that assembly.

H. Check Bit Memory Logic

The check bit memory circuit shown in the upper portion of FIGURE 19 is quite similar to the standard memory circuits (see the memory circuit description above). It is located on a plugboard above the left side of the Varipak assembly, in socket C. It is connected to memory position 21, and there is a bias circuit and bias head associated with it, although not shown in the figure (refer to FIGURE 18). The only difference between this circuit and the others is that only one timing pulse input is provided (at pin H). The logic is such that if the input G is at the "zero" level, the circuit recirculates the information contained on the drum, but if the input G is a "one," a "one" is then written on the drum. Thus once a "one" has been recorded in a given position on this track, it will remain there until the memory is erased.

I. Timing Logic

The timing circuit shown in the lower portion of FIGURE 19 is contained on a plugboard located above the center of the Varipak assembly, in socket T. The timing pulses are obtained from one of the two "DP"

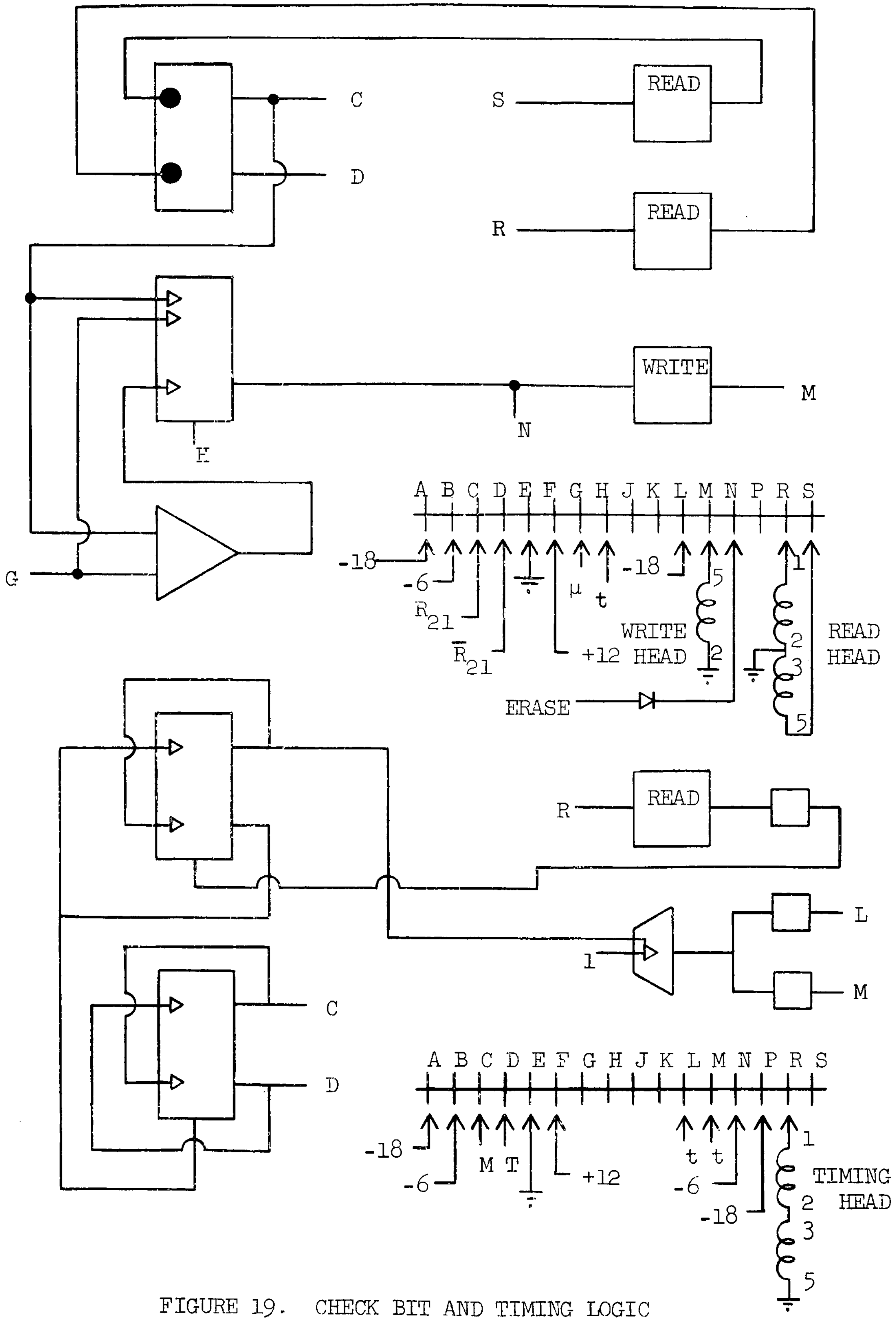


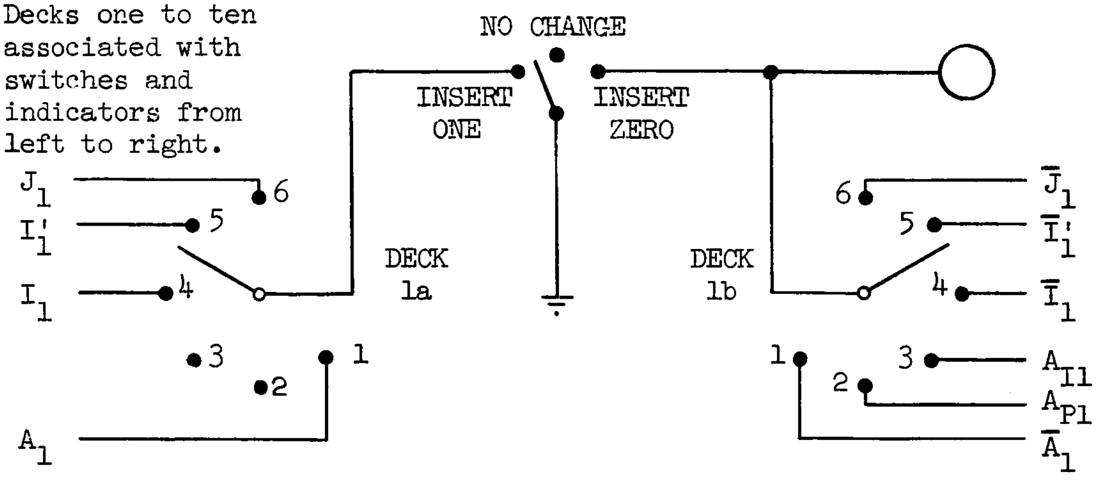
FIGURE 19. CHECK BIT AND TIMING LOGIC

timing tracks located on the drum, and have a frequency of 125 Kilocycles. Refer to Appendix C for a method of recording other timing tracks on the drum. Because the adjustment of the read and write heads in the machine is quite critical at this frequency, the timing circuit divides the input frequency and supplies pulses to the machine at a 62.5 Kilocycle rate.

There is only one read circuit, and there are no write circuits associated with this card. Note that the timing head is not grounded at its center, and therefore produces pulses of approximately double the magnitude of the average read pulse. The read circuit is adjusted to produce short, flat-topped, positive-going pulses, which are powered by an inverter circuit and used as the AC input to a flip-flop (note this is not a read flip-flop). This flip-flop provides frequency division, and its "one" output is shaped by a pulse gate, powered by two inverters, and supplied as machine clock pulses (t) on pins L and M. The "zero" output of the flip-flop is used to drive the AC inputs of a second flip-flop. This second flip-flop produces the alternating clock pulses M and T (available at pins C and D) which are used in the analog-to-digital converter. Several of the pins shown as blanks in the figure actually are connected to various points in the timing circuit which are not used in the machine. It must be remembered that the pin designations shown in FIGURE 19 have no relation to system signals designated by the same letters.

J. Check Circuit

The check circuit of FIGURE 20 consists of a 10 deck rotary switch, ten "bit-insertion" switches, and ten indicator lamps and drivers, all located on the front panel.


Decks one to ten associated with switches and indicators from left to right.

	POSITION					
	1	2	3	4	5	6
DECK	SHIFT REGISTER	PREDICT	IDENTIFY	PREDICT DAC	IDENTIFY DAC	ADC
1a	A_1	-	-	I_1	I'_1	J_1
1b	$\bar{A}_1$	A_{P1}	A_{I1}	$\bar{I}_1$	$\bar{I}'_1$	$\bar{J}_1$
2a	A_2	-	-	I_2	I'_2	J_2
2b	$\bar{A}_2$	A_{P2}	A_{I2}	$\bar{I}_2$	$\bar{I}'_2$	$\bar{J}_2$
3a	A_3	-	-	I_3	I'_3	J_3
3b	$\bar{A}_3$	A_{P3}	A_{I3}	$\bar{I}_3$	$\bar{I}'_3$	$\bar{J}_3$
4a	A_4	-	-	I_4	I'_4	J_4
4b	$\bar{A}_4$	A_{P4}	A_{I4}	$\bar{I}_4$	$\bar{I}'_4$	$\bar{J}_4$
5a	A_5	-	-	I_5	I'_5	J_5
5b	$\bar{A}_5$	A_{P5}	A_{I5}	$\bar{I}_5$	$\bar{I}'_5$	$\bar{J}_5$
6a	A_6	-	-	I_6	I'_6	J_6
6b	$\bar{A}_6$	A_{P6}	A_{I6}	$\bar{I}_6$	$\bar{I}'_6$	$\bar{J}_6$
7a	A_7	-	-	G_4	G_4	G_4
7b	$\bar{A}_7$	A_{P7}	A_{I7}	$\bar{G}_4$	$\bar{G}_4$	$\bar{G}_4$
8a	A_8	-	-	G_3	G_3	G_3
8b	$\bar{A}_8$	A_{P8}	A_{I8}	$\bar{G}_3$	$\bar{G}_3$	$\bar{G}_3$
9a	A_9	-	-	G_2	G_2	G_2
9b	$\bar{A}_9$	A_{P9}	A_{I9}	$\bar{G}_2$	$\bar{G}_2$	$\bar{G}_2$
10a	A_{10}	-	-	G_1	G_1	G_1
10b	$\bar{A}_{10}$	A_{P10}	A_{I10}	$\bar{G}_1$	$\bar{G}_1$	$\bar{G}_1$

FIGURE 20. CHECK CIRCUIT

The "register select" switch may be set to connect the "bit-insertion" switches and the indicators to the following points:

1) SHIFT REGISTER — The first ten shift register bits (A_1 through A_{10})

2) PREDICT * — The input to coincidence circuit No. 1 (A_{P1} through A_{P10})

3) IDENTIFY * — The input to coincidence circuit No. 2 (A_{I1} through A_{I10})

4) PREDICT DAC — The digital-to-analog converter No. 1 (I_1 through I_6)
The weighting register (G_4 through G_1)

5) IDENTIFY DAC — The digital-to-analog converter No. 2 (I'_1 through I'_6)
The weighting register (G_4 through G_1)

6) ADC — The analog-to-digital converter (J_1 through J_6)
The weighting register (G_4 through G_1)

* These circuits register the complement of each bit observed. See text below.

Each of the ten bits may be observed on its respective indicator light (a lamp on indicates a "one"), and the "bit-insertion" switches may be used to clamp any particular flip-flop to either the "one" or the "zero" state. The "predict" and "identify" positions are special cases in that a light on represents a "zero" rather than a " one," and the "bit-insertion" switches can not be used in these positions. Care should be taken not to change the position of the "register select" switch during normal operation, since the transients generated may destroy the information in various registers. It is also necessary that all of the "bit-insertion" switches be in the "no change" position when running the model, to prevent the insertion of false information.

PART V.

PHYSICAL DESCRIPTION

The entire machine, not including the external clock and the analog computer, is contained in a cabinet 51 inches high by 28 inches wide by 28 inches deep. The upper front side of this cabinet is sloped to accomodate the front panel. The memory drum is mounted on the floor of the unit, and the DC power supply is suspended just below the top of the cabinet. The interior of the unit is accessible from a back door and from two hinged side panels (fastened shut with machine screws). For mobility, the entire cabinet is mounted on wheels.

All logic circuits except the two circuit boards purchased from the Computer Control Company and used in the analog-to-digital converter are mounted on "Vectorboard" epoxy paper circuit boards. Those circuits related to the drum memory are contained on twenty-two 4.5 by 6.5 inch plugboards which are mounted in an ELCO Varipak assembly (two of the boards are actually mounted above the Varipak assembly). These boards plug into mating sockets and may easily be removed for servicing. All other circuits are mounted on 4.75 by 17 inch boards which are held in a rack above the memory drum. These boards are permanently wired in, but each board may be lifted above the level of the other boards in the rack for testing and repairs.

From right to left as seen from the rear of the machine, the large circuit boards contain the following logic circuits:

1) Shift Register Logic, Part 1
2) Shift Register Logic, Part 2
3) Coincidence logic No. 1
4) Coincidence logic No. 2

5) Timing and Control Logic, Part 1
6) Timing and Control Logic, Part 2
7) Weighting Logic
8) Weighting Relays and Resistor Networks (this circuit is not mounted on a standard Vectorboard)
9) Digital-to-Analog Converter No. 1
10) Digital-to-Analog Converter No. 2
11) Analog-to-Digital Converter Logic

At the far left side of the rack, the two 3C cards are located in their respective sockets.

FIGURE 21, FIGURE 22, and FIGURE 23 are three photographs of the machine taken during its construction. FIGURE 21 shows the front panel of the device. At the upper right of this panel are the input-output connections: external clock, binary input, and a socket for connecting the machine to an analog computer (a mating connector terminating in color-coded banana plugs is supplied for this purpose).

FIGURE 22 is a view of the right side of the machine with the door panel opened. From top to bottom, one can observe:

1) The DC power supply.
2) The two 3C circuit boards, and behind them the 4.75 by 17 inch logic boards.
3) The rear of the ELCO Varipak assembly.
4) The memory drum assembly (with cover removed), showing the tops of the read-write heads and the plug and socket arrangement associated with each head.
5) The power control chassis.

FIGURE 23 is a view of the left side of the machine with the door panel opened. From top to bottom, one can observe:

1) The DC power supply.
2) The check bit and timing circuit memory cards.
3) The ELCO Varipak assembly with several cards removed to show the guides and sockets.
4) The memory drum assembly and mist oiling apparatus.
5) The bias circuits.

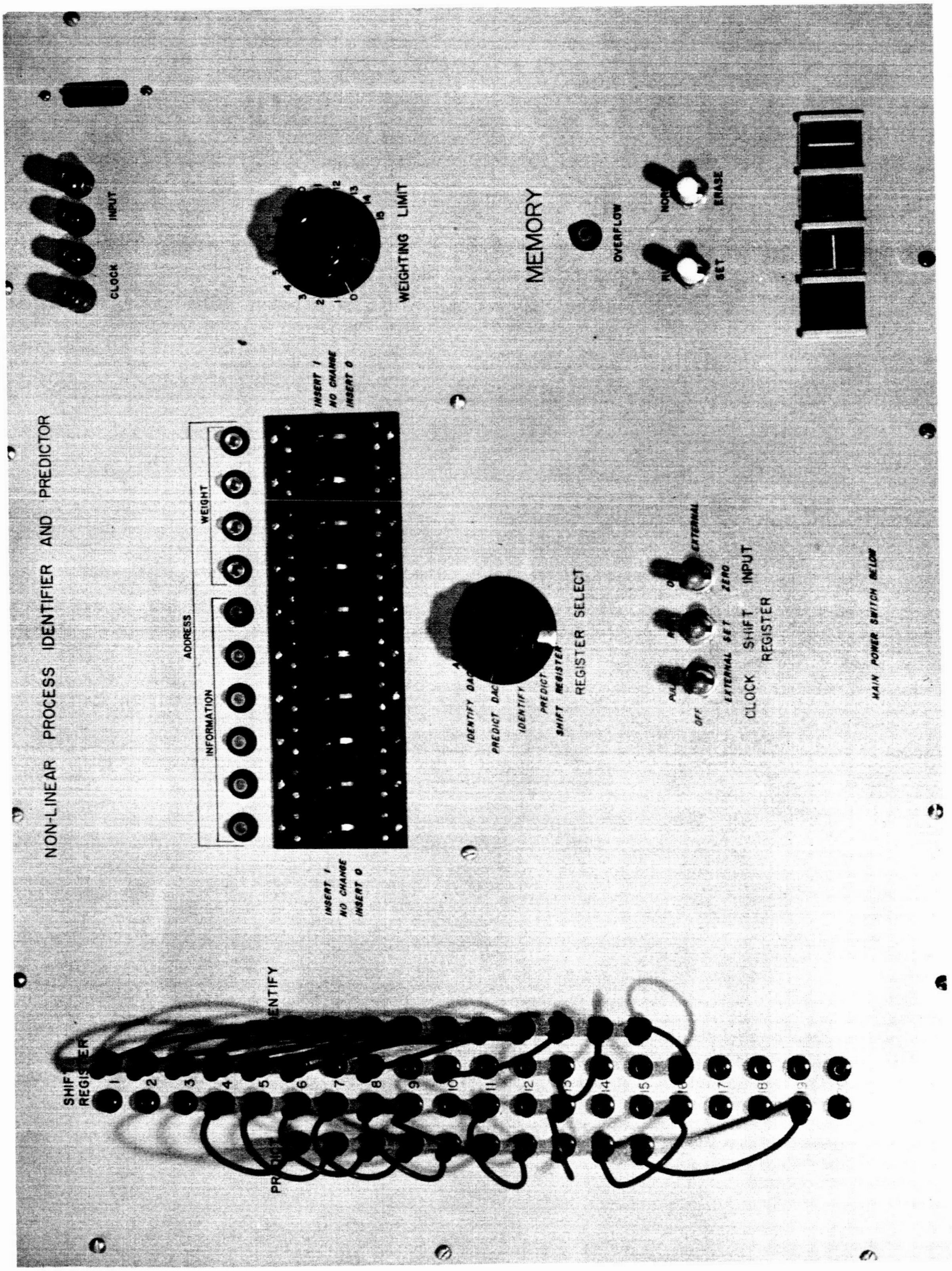

NON-LINEAR PROCESS IDENTIFIER AND PREDICTOR
CLOCK
INPUT
WEIGHTING LIMIT
0 1 2 3 4 5 ... 12 13 14 15
MEMORY
OVERFLOW
SET
ERASE
ADDRESS
INFORMATION
WEIGHT
INSERT 1
NO CHANGE
INSERT 0
INSERT 1
NO CHANGE
INSERT 0
IDENTIFY DAC
PREDICT DAC
IDENTIFY
PREDICT
SHIFT REGISTER
REGISTER SELECT
OFF
EXTERNAL
CLOCK
SET
ZERO
EXTERNAL
SHIFT REGISTER INPUT
MAIN POWER SWITCH BELOW
SHIFT REGISTER
IDENTIFY
PREDICT
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Figure 21.

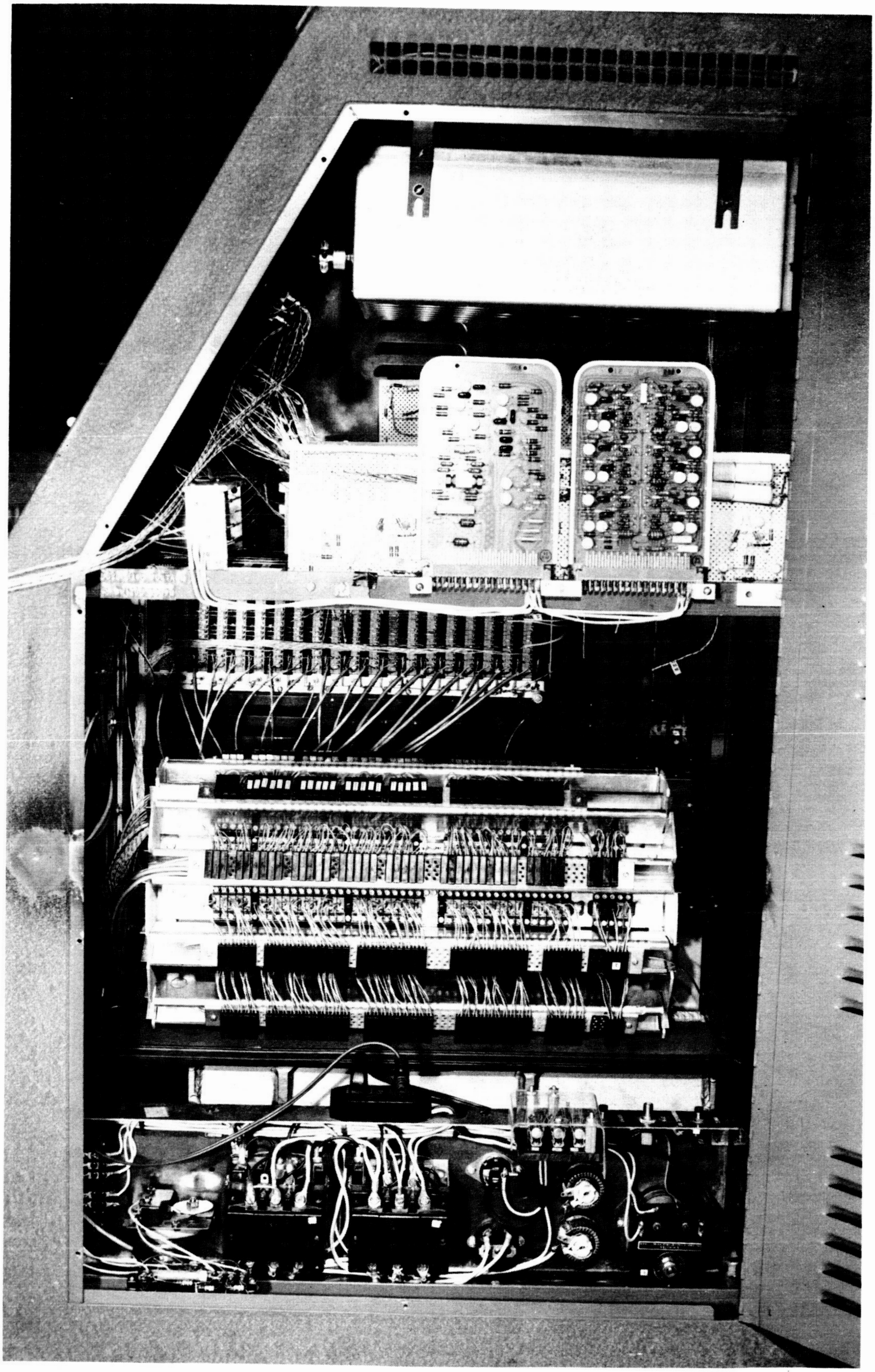


Figure 22.

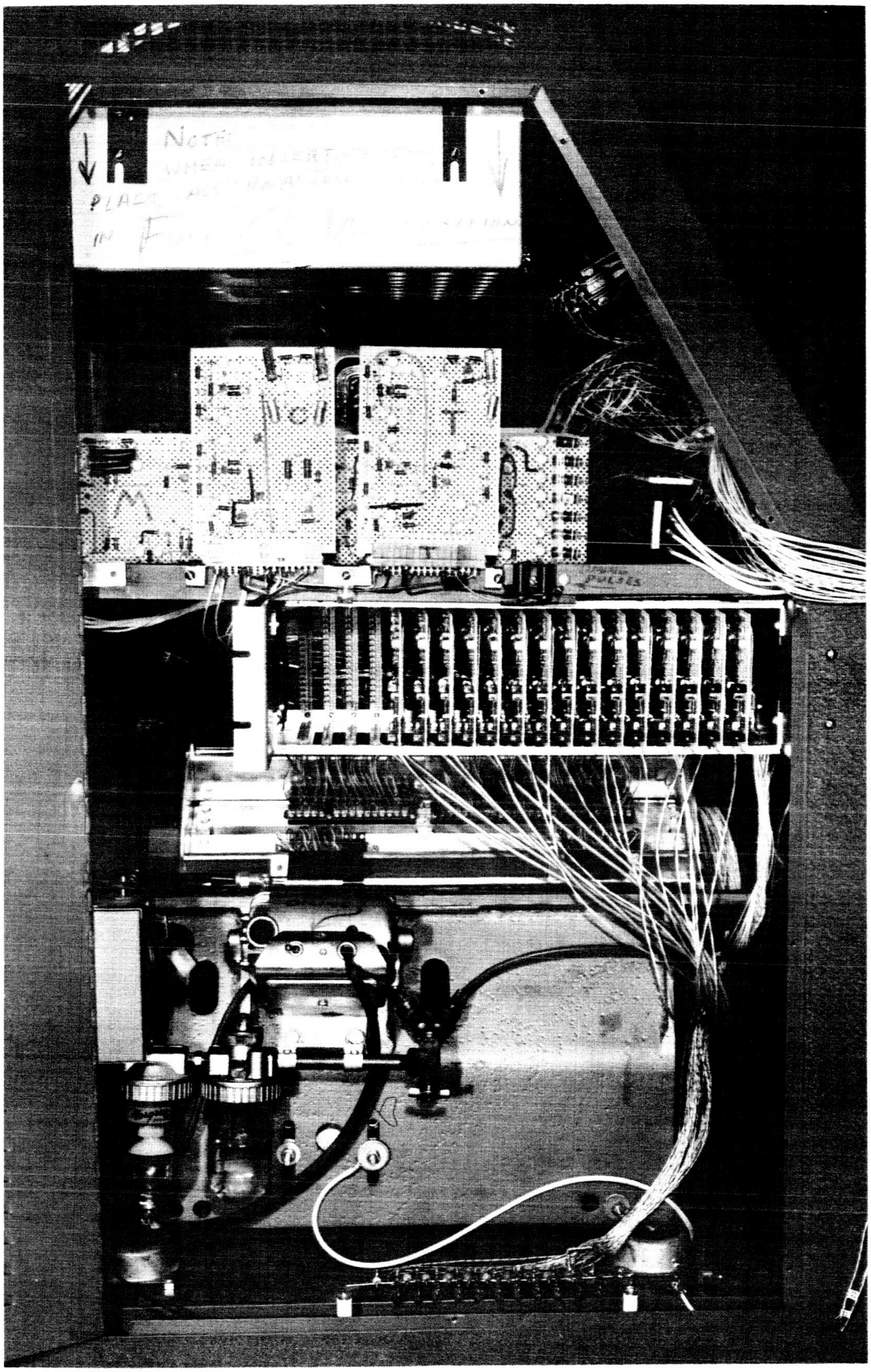


Figure 23.

PART VI.

INTERNAL ADJUSTMENTS

A. Memory Test

Because the memory circuits are the most sensitive to heating effects and changes in components due to aging, it is logical to test the entire memory whenever malfunctions occur which can not be directly traced to some other point in the circuitry.

A simple but quite thorough memory test may be performed with the aid of an oscilloscope. First open the left side door panel so the Varipak assembly and memory cards are accessible. Rotate the "register select" switch on the front panel to the "ADC" position and place all ten "bit-insertion" switches in the "insert one" position. Check to make sure all patch cords are inserted and then place the input switch in the "one" position. Either manually or with an external clock provide at least 20 clock pulses (to fill the shift register with "one" bits).

Next place the clock switch in the "off" position, erase the memory, and manually supply one clock pulse to the system. If the memory is working properly, each position (1 through 21) should now contain all "zeros" except for a single "one." If the "one" side of each read flip-flop is now observed with the oscilloscope, a positive pulse (from -6 volts to ground, 16 microseconds wide) will be observed every 3.85 milliseconds. Any other pattern would indicate a fault in the memory position being observed. The "one" side of the read flip-flop may be checked by observing the collector of the uppermost transistor on the top edge of the memory card. This point is accessible without removing any cards from the Varipak assembly.

If the "one" side of the check bit read flip-flop (position 21, located at the left above the Varipak) is used to trigger the oscilloscope, the pulses from positions 11 through 20 will appear to be delayed by 16 microseconds, and the pulses from positions 1 through 10 will appear to be delayed by 32 microseconds.

B. Memory Adjustment

Adjustment of a memory card is performed by the use of the special memory test card. This card is inserted between the memory card to be adjusted and the corresponding socket in the Varipak. The test card acts as an extender, connecting the card to be tested to its socket, and also supplies a constant 101010 input to the write flip-flop. The test card must never be used in the C or T sockets above the Varipak. The cable and plug from the test card must be connected to a source of timing pulses (positive-going, 6 volt pulses). For final adjustment of a memory card, this plug should be inserted in the socket marked "timing pulses" which is mounted just above the Varipak. Make sure that the ground shield of the cable is connected to the ground (black wire) side of the socket to prevent damage to the timing circuits.

Turn both potentiometers on the card to be adjusted to the counterclockwise position, insert the test card in the appropriate socket, insert the card to be adjusted in the test card socket, and turn on the DC power and the external pulse generator, if used. Check the write flip-flop and the write circuit to make sure that they are operating properly. Now observing each collector of the two read circuits in turn (using an oscilloscope), turn the corresponding potentiometer in a clockwise direction until a 6 volt pulse (from -6 volts to ground) is obtained. Each circuit should be adjusted to the point where the output has a clearly defined flat top. If such an output can not be obtained, there is a fault in the read circuit, the read head, the bias circuit, or the bias head (assuming that the write circuit has already been checked and found to be operating properly).

Next observe the "one" side of the read flip-flop and make any further small adjustments of the potentiometers which tend to make this output more stable. A small noise spike will be present on the output, caused by the system clock pulses. It is useful when making final adjustments of a read circuit to remember that the read flip-flop output is to be sampled at these clock-pulse times.

The timing card receives a steady stream of pulses from the timing head, and thus no test card is needed when adjusting its read circuit.

The check bit circuit is difficult to adjust since the test card can not be used in this position. If the nor circuit input from the "one" side of the read flip-flop is temporarily disconnected, the check bit card may be adjusted by using it in any of the other memory positions (1 through 20) and making use of the test card as above. After adjustment, remember to reconnect the nor circuit input, and then make any final adjustments by observing the card in actual operation in its usual position.

C. Digital-to-Analog Converter Adjustment

To adjust the digital-to-analog converter No. 2, connect analog point M (dark green plug) to the -10 volt reference, and connect analog point N (brown plug) to an amplifier with a 10,000 ohm feedback resistor. Set the "register select" switch on the front panel to "identify DAC," and use the "bit-insertion" switches to place the test numbers in the information portion of the register. Observe the output of the amplifier and adjust to the voltage specified, using the six potentiometers on the DAC No. 2 circuit board.

TEST NUMBER	ADJUST	VOLTAGE
1 0 0 0 0 0	No. 1 (rear)	5.00
0 1 0 0 0 0	No. 2	2.50
0 0 1 0 0 0	No. 3	1.25
0 0 0 1 0 0	No. 4	0.625
0 0 0 0 1 0	No. 5	0.312
0 0 0 0 0 1	No. 6 (front)	0.156

To adjust digital-to-analog converter No. 1, proceed as before with analog point L (red plug) connected to the +10 volt reference, and analog point H (violet plug) connected to the amplifier. Place the "register select" switch in the "predict DAC" position and adjust the potentiometers on the DAC No. 1 circuit board. Note that the observed voltages will all be negative for this case.

D. Weighting Circuit Adjustment

Make the following analog connections: point A (yellow plug) to an amplifier input, point B (orange plug) to the amplifier output, and point P (white plug) to the -10 volt reference. There is no amplifier feedback resistor used, and no other plugs should be connected.

Using the "register select" switch and the "bit-insertion" switches, place the following test numbers in the weighting register and adjust for the specified voltage at the amplifier output, using the eight potentiometers on the relay board (the relay board is mounted in the circuit board rack, next to the weighting circuit board).

TEST NUMBER	ADJUST	VOLTAGE
0 0 0 0	- (check only)	10.00
0 0 0 1	$\overline{A}$	5.00
0 0 1 0	$\overline{B}$	3.33
0 1 0 0	$\overline{C}$	2.00
1 0 0 0	$\overline{E}$	1.11

Now disconnect point P (white plug), and connect point F (blue plug) to the -10 volt reference.

TEST NUMBER	ADJUST	VOLTAGE
0 0 0 0	- (check only)	0.00
0 0 0 1	A	5.00
0 0 1 0	B	6.66
0 1 0 0	C	8.00
1 0 0 0	E	8.89

The potentiometers on the relay board are clearly labeled with the letters used above.

PART VII.

OPERATING PROCEDURE

To operate the model, first make the analog connections shown in FIGURE 2. Use an attenuator at the process output connection, and select an RC combination for the process averager such that RC=T (approx.), where T is the period of the input clock which is to be used. Connect the source of input clock pulses (6 volt, positive-going) to the machine and adjust it to the proper frequency if necessary. Connect the system to be observed to the attenuator mentioned above, set the attenuator to zero, and connect the system binary input (±5 to ±50 volts) to the appropriate terminals on the machine.

Turn on the machine, and after the drum has reached running speed ("start" indicator comes on), turn on the machine DC power and the analog computer power at the same time. This measure prevents excessive current from flowing between the two units.

Check all reference voltages (+10, -10, +5, and -5), and adjust them to the proper values. On the machine, set the weighting limit switch to zero, place the clock switch in the "external" position, place the shift register set switch in the "set" position, and turn the register select switch to the "ADC" position (it is often useful to leave the register select switch in this position during the operation of the machine). Make sure that all of the "bit-insertion" switches are in the "no change" position.

If the amplifier associated with the process averager has overloaded, use a jumper or a low value feedback resistor to bring it back into operating range. It should now no longer overload, and should have a zero volt output (since the attenuator at its input is at zero). Erase the machine memory and observe the indicator lights. The binary number 32 (which represents a zero input) should be in the register. A one bit difference (binary 31 or 33) is of no importance, representing only a slight offset in the comparator unit of the analog-to-digital converter.

Now force the observed system to its maximum positive output and slowly adjust the attenuator at the process averager input until a binary number between 60 and 62 appears in the register. Next force the system to its maximum negative output and observe the number in the information portion of the register. If this number is zero, decrease the attenuator setting until a number between 1 and 3 is observed. The model has now been adjusted for its maximum operating range.

Place the shift register set switch in the "run" position, set the weighting limit switch to the desired limit, make sure that the clock switch is at "external," the memory switch is at "run," and the input switch is at "external."

Place the patch cords from the "identify" side of the patch board for the desired taper along the shift register, starting at position 1. Place the patch cords from the "predict" side of the patch board in exactly the same taper configuration as is used on the "identify" side. If a predicted output is desired, start this taper at position 2, 3, 4, etc. The number of blank positions before the first "predict" patch point will be equal to the number of input clock bits of prediction.

Prediction time will equal the number of blank positions times the input clock period, or $\frac{\text{blank positions}}{\text{input clock frequency}}$ = prediction time.

Erase the memory to remove unwanted information, and proceed with normal machine operation. Remember that any change of the weighting limit switch or register select switch will cause transients and should be followed by an erase operation. If the input clock speed is changed, the attenuator at the process output must be adjusted accordingly. Large variations in clock frequency will necessitate a change in the process averager RC combination. If at any time the input clock is turned off, the amplifier associated with the process averager will overload and must be manually reset to zero as before.

If it is desired to turn off the system being observed, the memory set switch must first be placed in the "set" position or the model will immediately start to adapt to the turned-off system. This switch may be used at any time that it is wished to disconnect the model from the system without destroying the model memory.

PART VIII.

TESTS AND RESULTS

A. Test Systems Used

In order to check the capabilities of the model, several test systems were programmed on the unused portion of the PACE TR-10 analog computer (a portion of this machine was used as the analog section of the model). The systems used are shown in FIGURE 24. Input to all systems was a pseudo-random binary function generated by a Digital Electronics Corporation DIGIAC 3010 Digital Logic Trainer and a General Radio Company Unit Pulse Generator (refer to FIGURE 32). The high-frequency clock of the DIGIAC (about 10 Kilocycles) drives a flip-flop which is sampled at intervals of 2, 4, or 8 model input clock pulses (supplied by the Unit Pulse Generator). Relative drift in the two timing devices causes a pseudo-random rather than a repetitive binary function to be formed. Astable multivibrators are used for delaying and shaping of pulses, inverters are used to provide output drive.

A four channel Sanborn recorder was used to observe the input, the systems, and the model. It should be noted that the recorder channel used to observe the random binary input suffered an amplifier failure early in the test series. Since the output of this channel was still sufficient to show input switching times, it was decided to ignore the failure and proceed with the tests. All binary inputs were between the levels of +10 and -10 volts throughout the test. Attenuators were used to reduce the system outputs to ±5 volts.

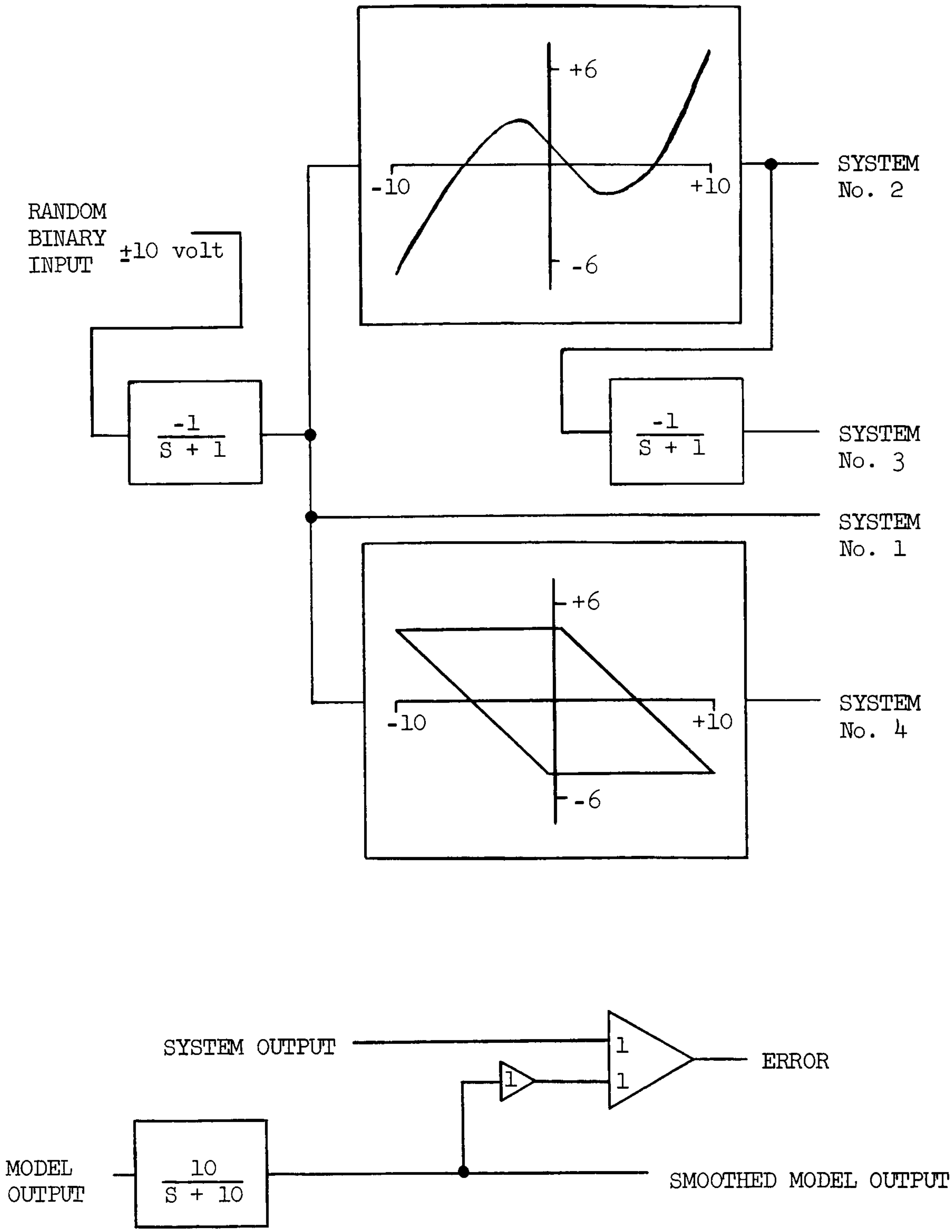


FIGURE 24. TEST SYSTEMS

B. Results

All of the graphs shown were made after the model had been observing the system in question for between one and three minutes (this parameter was not measured during the tests). This is quite consistant with Roy and DeRusso's predicted identification time of 14 to 70 times the settling time of the system. [6]

FIGURE 25 shows the results of modeling a simple $\frac{-1}{S + 1}$ system (system No. 1). An input clock rate of 2.5 pulses per second was used, with a minimum of 4 input clock pulses per binary input change. An untapered delay line was used, and the weighting limit was 15. It can be seen that smoothing the model output (through a $\frac{10}{S + 10}$ transfer function) helps create a better looking output, but induces a slight delay in the model output.

FIGURE 26 shows the result of modeling the more complex system formed by following the $\frac{-1}{S + 1}$ function with a non-linear element (system No. 2, FIGURE 24). The same input conditions and machine limits as above were used. Note that the model has difficulty in identifying the fast, low amplitude portion of the system output, and tends to give an average value for this portion of the output.

FIGURE 27 and FIGURE 28 show the use of prediction to compensate for the delay caused in the model output smoothing circuit. The system observed is composed of a $\frac{-1}{S + 1}$ function and a non-linear element followed by a second $\frac{-1}{S + 1}$ function (system No. 3). A 2.5 pulse per second input clock was used with a minimum of 8 clock pulses per binary input change. An untapered line and a weighting limit of 15 were employed. In FIGURE 27 the model is identifying the system

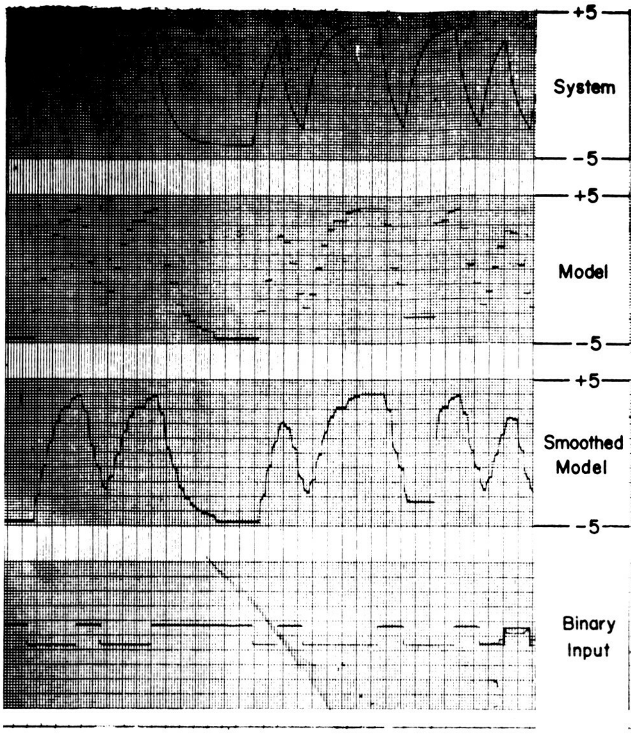


FIGURE 25 - IDENTIFICATION SYSTEM No. I

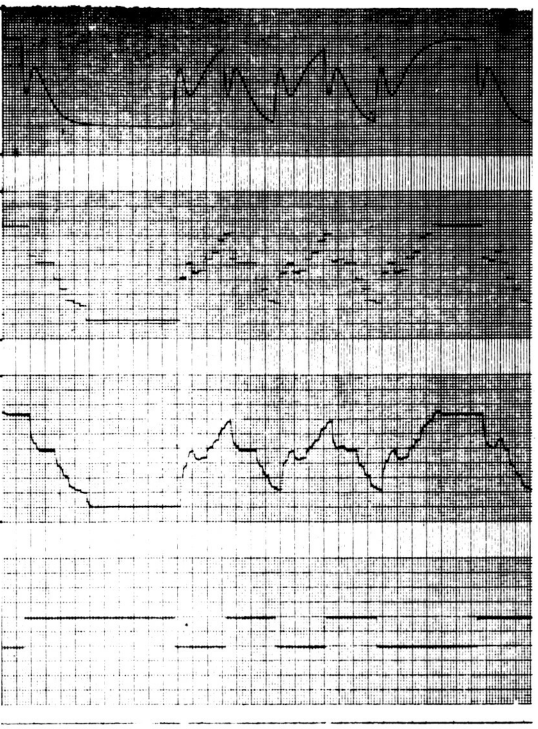


FIGURE 26 - IDENTIFICATION SYSTEM No. 2

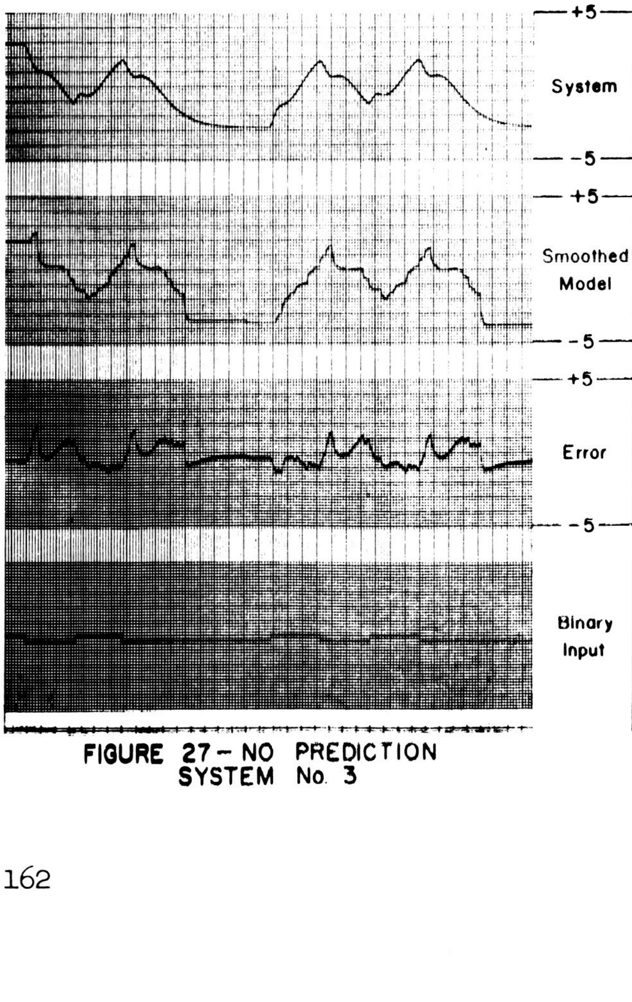


FIGURE 27 - NO PREDICTION SYSTEM No. 3

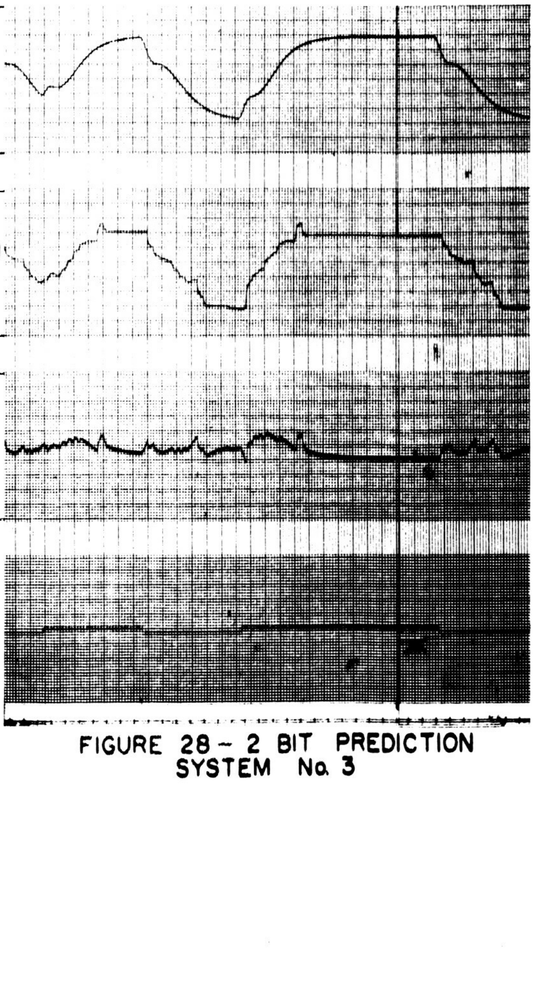


FIGURE 28 - 2 BIT PREDICTION SYSTEM No. 3

quite well, but a rather large error is present due to the delay in the model smoothing circuits. In FIGURE 28 a 2-bit (input clock bit-time) prediction is used to compensate for this delay, and the error in this case is much smaller than before, being caused primarily by fast variations in the system which are represented by an average value at the model output. The actual prediction time used in this example is 0.8 seconds.

FIGURE 29 illustrates the action of the model in identifying the $\frac{-1}{S + 1}$ transfer function (system No. 1) when the input binary function is allowed to change every 2 input clock pulses. An untapered delay line, weighting limit of 15, and 2.5 pulse per second input clock were used in this test. A one bit prediction is used to offset the delay caused by model output smoothing. The wide bandwidth of this system makes it difficult to model, and the comparatively rapid frequency of input changes adds further to the difficulty of obtaining a small error in this model.

FIGURE 30 gives the results of modeling a system with a large amount of hysterisis (system No. 4, FIGURE 24). A 2.5 pulse per second input clock is used with a minimum of 4 bits per binary input change. A prediction of one bit is used to compensate for smoothing delay, weighting limit is 15, and the delay line is untapered.

FIGURE 31 illustrates the use of the machine as an adaptive model. The system observed in this case was abruptly changed from system No. 1 to system No. 3. These systems differ not only in general output waveforms, but are also of opposite polarity for a given binary input. Thus a true "worst case" was presented to the model. An input

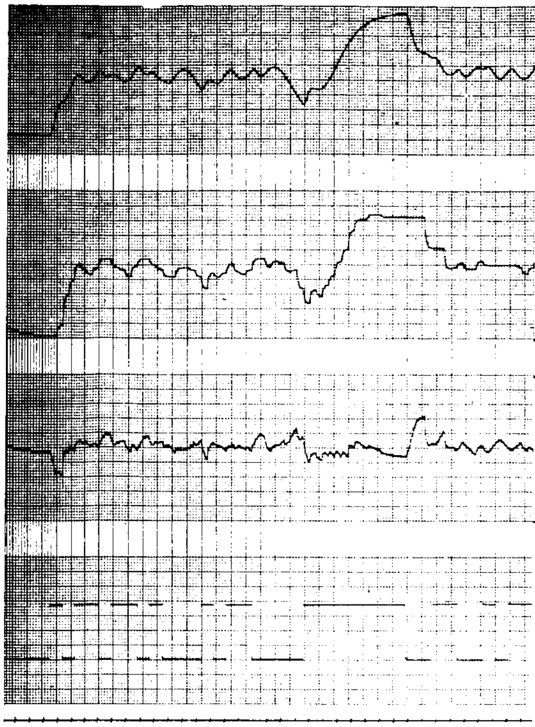


FIGURE 29 — MODELING SYSTEM No. 1

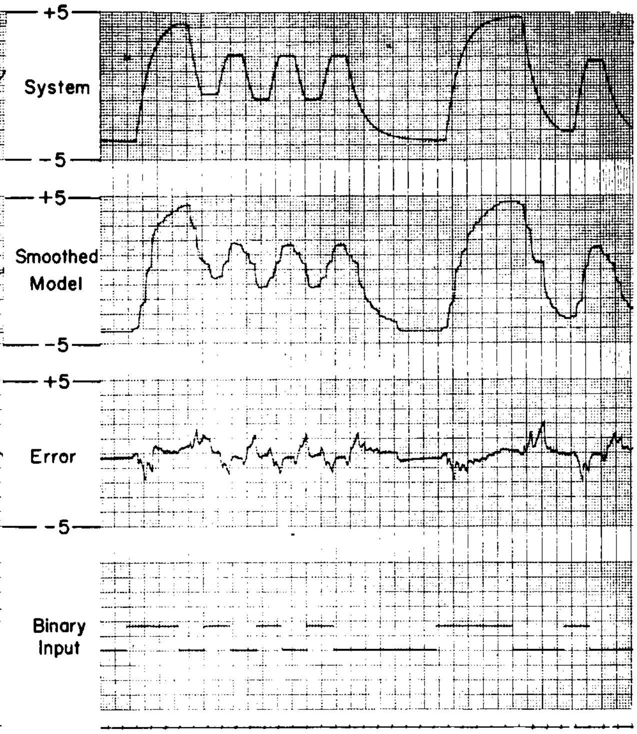


FIGURE 30 — MODELING SYSTEM No. 4

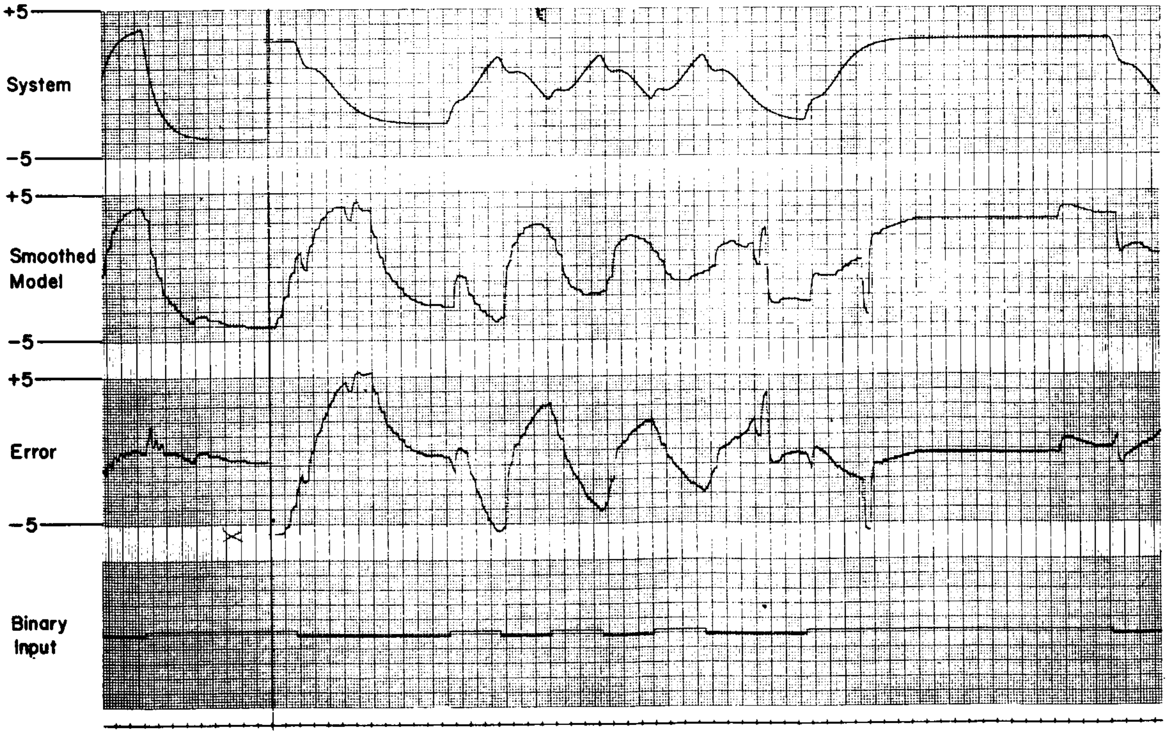


FIGURE 31 — ADAPTIVE MODEL

clock frequency of 2.5 pulses per second was used, with a minimum of 8 bits per binary input change. An untapered delay line was used with a 1 bit prediction to compensate for smoothing delay. A weighting limit of 2 was used so that the model could adapt quickly to the changing system. Although the error does not reach a minimum in the short record shown, it is obvious that the model is adapting very quickly indeed to the abruptly changing system.

A test was made using a tapered input delay line, but no significant improvement in modeling was observed. This may be due in part to the nature of the systems being observed (systems No. 1 and No. 3), and in part to the fact that the 20 bit shift register allows only rather approximate tapers to be set up. In another test, a prediction of 4 bits or more was attempted, but because of the large magnitude of the initial response of a system containing $\frac{1}{S + 1}$ elements, the results were rather poor. The prediction which was used to compensate for model smoothing delays serves to illustrate the model's predictive capabilities, and it seems reasonable to assume that given a system with a smaller bandwidth this prediction could be extended over a much greater range.

It should be noted that the pseudo-random binary input used was far from optimum because of the possibility that certain patterns would tend to repeat. Even more important is that when the minimum number of input clock pulses per binary input change was set to 4 (for example), this limited the binary function to changing at intervals of 4 (ie. 4, 8, 12, 16, etc.) while a true random input would be able to change at any time after 4 intervals (ie. 4, 5, 6, 7, 8, etc.).

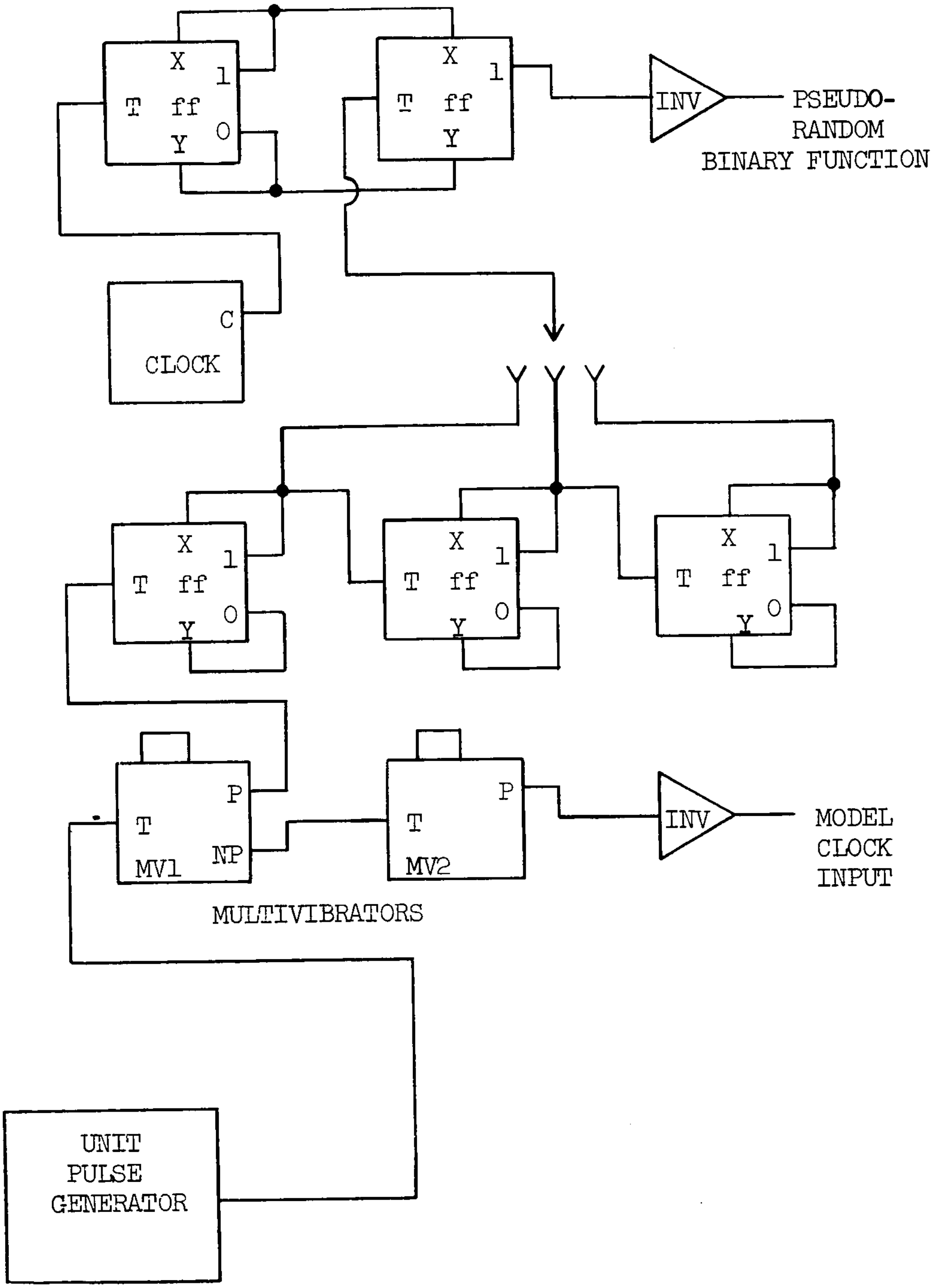


FIGURE 32. GENERATION OF A PSEUDO-RANDOM BINARY FUNCTION ON THE DIGIAC 3010

PART IX.

LITERATURE CITED

1. Computer Control Company, Inc. Instruction Manual for S-Pac Digital Modules. Publications No. 71-100A. Framingham, Mass., April 1962.

2. International Business Machines Corp. IBM Customer Engineering Reference Manual, Preventive Maintenance and Adjustments, IBM 650 Data Processing System. Form 22-6316-0. New York, 1957.

3. Jackson, Albert S. Analog Computation. New York, McGraw-Hill, 1960. p. 565-567.

4. Lohan, Frank J. Introduction to Pulse Magnetic Recording. Publication B 6023. Walled Lake, Mich., Bryant Computer Products.

5. Roy, R. A Digital Orthogonal Model for Nonlinear Processes. Doctoral Thesis. Rensselaer Polytechnic Institute, November 1961.

6. Roy, R. and DeRusso, P. M. A Digital Orthogonal Model for Nonlinear Processes with Two-level Inputs. IRE Trans. Professional Group on Automatic Control, October 1962, p. 93-101.

7. Roy, R., Miller, R. W. and DeRusso, P. M. An Adaptive-Predictive Model for Nonlinear Processes with Two-level Inputs. Fourth Joint Automatic Control Conference, Paper VIII-3, p. 204-210, June 1963.

8. Ware, Willis W. Digital Computer Technology and Design, Vol. II: Circuits and Machine Design. New York, John Wiley and Sons, 1963.

PART X.

APPENDIXES

A. Input-Output Specifications

AC INPUT

2 lines and neutral, 60 cps.
208-230 volts line to line
115 volts line to neutral
power consumption 10 amps per line

MINIMUM AMBIENT OPERATING TEMPERATURE

68 degrees F.

CLOCK INPUT

6 volt positive-going pulses
maximum frequency 75 pulses per second

MODEL INPUT

binary signal ±5 to ±50 volts

MODEL OUTPUT

±5 volts maximum
64 levels, 0.156 volts per level

ANALOG COMPUTER REQUIRED

four operational amplifiers capable of ±10 volt output
reference voltages of +10, +5, -5, and -10 volts
model originally designed for use with EAI PACE TR-10 computer

INTERNAL DC POWER

Computer Control Company model RP-31 power supply
input 115 volts, 60cps.
regulated output -18 volts, 8 amperes
- 6 volts, 8 amperes
+12 volts, 1.6 amperes

MACHINE CLOCK SPEED

62.5 Kc.

NOMINAL MEMORY DRUM SPEED

12,500 rpm.
3.85 milliseconds from write head to read head

DRUM BEARING LUBRICATION

10 seconds of mist oiling every 24 hours of operation
must be oiled from rear of cabinet--front panel controls do not control lubrication of drum.

B. Wire Designations

All signal-carrying wires in the machine are marked with numbered tags at both ends except for those going to the 10 deck "register select" switch and those going to the patch board. These two groups are left unmarked. There are also no markers on power lines, or on the connections between the memory logic and the read, write, and erase heads.

The following is a list of wire designations by circuit board, giving the signal carried and the point of termination of each wire:

SIGNAL	TERMINATION	WIRE NUMBER
Shift Register No. 1		
S.	shift 2	99
t	timing	88
H	shift 2	90
input	panel	89
$\overline{K}$	shift 2	94
$\overline{K}$	T and C 2	19
$\overline{K}$	ADC	77
A_{10}	shift 2	96
$\overline{A}_{10}$	shift 2	95
Coincidence No. 1		
C.	T and C 1	4
R_1-R_{10}	coin 2	150-159
A_{P1}-A_{P10}	memory	140-149
Coincidence No. 2		
C'	T and C 1	1
R_1-R_{10}	coin 1	150-159
R_1-R_{10}	memory	160-169

SIGNAL	TERMINATION	WIRE NUMBER
Timing and Control No. 1		
C'	coin 2	1
R_{21}	memory	2
$\overline{R}_{21}$	memory	3
C.	coin 1	4
t	timing	5
K	shift 2	6
$\overline{X}$	T and C 2	7
t	timing	8
Y	DAC 2	9
$\overline{V}$	T and C 2	10
W	T and C 2	11
ω	memory	13
α	memory	15
α	T and C 2	16
π	memory	17
ϵ	memory	18
ϵ	T and C 2	14
$\overline{\mu}'$	T and C 2	12

SIGNAL	TERMINATION	WIRE NUMBER
Shift Register No. 2		
E	weight	97
t	timing	92
H	shift 2	90
H	panel	91
input	panel	93
$\bar{K}$	shift 1	94
K	weight	31
K	T and C 1	6
K	T and C 2	22
S.	shift 1	99
A_{10}	shift 1	96
$\bar{A}_{10}$	shift 1	95
Analog-to-Digital Converter		
$\bar{K}$	shift 1	77
F	T and C 2	25
T	timing	78
J_1-J_6	comparator	71-76
J_1-J_6	memory	81-86
Q	T and C 2	21
Q	weight	98
$\bar{\Theta}_1$-$\bar{\Theta}_4$	weight	133-136
Digital-to-Analog No. 1		
L	T and C 2	28
t	timing	45
t	DAC 2	58
R_{11}-R_{16}	memory	46-56 (even)
$\bar{R}_{11}$-$\bar{R}_{16}$	memory	47-57 (odd)
R_{11}-R_{16}	DAC 2	59-69 (odd)
$\bar{R}_{11}$-$\bar{R}_{16}$	DAC 2	60-70 (even)
β	weight	32

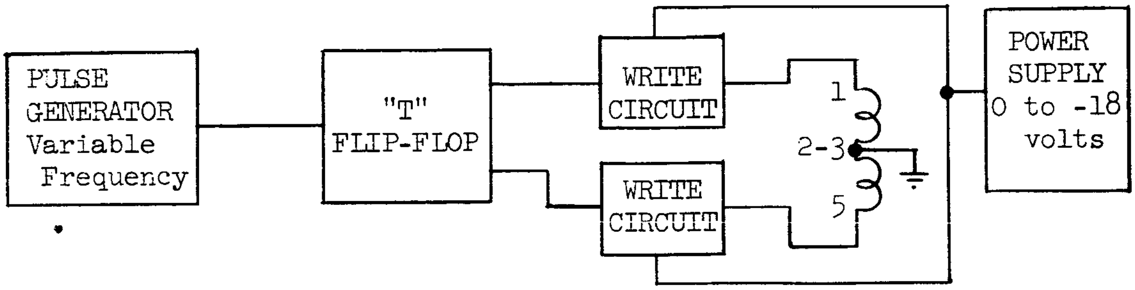
SIGNAL	TERMINATION	WIRE NUMBER
Timing and Control No. 2		
inhibit	panel	111
$\bar{K}$	shift 1	19
T	timing	20
W	T and C 1	11
$\bar{V}$	T and C 1	10
Q	ADC	21
α	T and C 1	16
ϵ	T and C 1	14
K	shift 2	22
t	timing	23
erase	panel	112
ovflow	panel	100
P	weight	24
F	ADC	25
F	weight	26
F	weight	27
L	DAC 1	28
μ	memory	29
$\bar{X}$	T and C 1	7
$\bar{\mu}'$	T and C 1	12
R_{21}	memory	87
indicator inputs from panel (right to left)		121-130
indicator outputs from panel (right to left)		101-110
Digital-to-Analog No. 2		
Y	T and C 1	9
t	DAC 1	58
R_{11}-R_{16}	DAC 1	59-69 (odd)
$\bar{R}_{11}$-$\bar{R}_{16}$	DAC 1	60-70 (even)

		WIRE
SIGNAL	TERMINATION	NUMBER
Weighting		
F	T and C 2	27
D	panel	131
K	shift 2	31
P	T and C 2	24
F	T and C 2	26
β	DAC 1	32
R_{17}-R_{20}	memory	33-39 (odd)
$\overline{R}_{17}$-$\overline{R}_{20}$	memory	34-40 (even)
$\overline{\Theta}_1$-$\overline{\Theta}_4$	ADC	133-136
G_1-G_4	panel	113-119 (odd)
$\overline{G}_1$-$\overline{G}_4$	panel	114-120 (even)
G_1-G_4	memory	41-44
E	shift 2	97

C. Method of Recording a Timing Track

If it becomes necessary (because of damage to both original timing tracks or because of a desire to change the basic clock frequency) to record a timing track, the following procedure may be used:

Select an unused memory track with two heads operating on it. Connect the output of one head to an oscilloscope, and connect the second head according to the diagram below.


PULSE GENERATOR Variable Frequency
"T" FLIP-FLOP
WRITE CIRCUIT
WRITE CIRCUIT
1
2-3
5
POWER SUPPLY 0 to -18 volts

The operation of the circuit is quite simple: At every pulse from the pulse generator the "T" flip-flop changes state, causing the direction of flux in the write head to reverse. As the frequency of the input pulse generator is varied slightly, the amplitude of the signal observed on the oscilloscope will vary depending on whether the signals recorded during each revolution of the drum are reinforcing or canceling the signals recorded during the previous revolution.

If the pulse generator is now set to twice the clock frequency desired, and then varied slightly to obtain a maximum output, the track in question will have recorded on it a frequency which is close to the desired frequency and also fits evenly on the drum (ie. it is constantly being reinforced). If the -18 volt source which supplies the write circuits is now slowly brought to zero (to avoid recording transients as it is shut off), the recording process will be completed. Disconnect the recording apparatus and note the recording head is no longer necessary.